Universidad
Carlos III de Madrid

TESIS DOCTORAL

The RBF-FD Method: Developments and Applications

Autor:
Víctor Bayona Revilla

Directores:
Manuel Kindelan Segura
Miguel Ángel Moscoso Castro

Instituto Gregorio Millán Barbany
Grupo de Modelización, Simulación Numérica y Matemática Industrial

Leganés, junio de 2013

TESIS DOCTORAL

The RBF-FD Method: Developments and Applications

Autor: Víctor Bayona Revilla

Directores: Manuel Kindelan Segura y Miguel Ángel Moscoso Castro

Firma del tribunal calificador:

(Nombre y Apellidos) Firma

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

A mis padres,

Agradecimientos

En primer lugar, me gustaría mostrar el más profundo agradecimiento a mis directores de tesis, Manuel Kindelan y Miguel Moscoso, por la confianza depositada en mí al proponerme este tema de investigación, así como por todo el apoyo y dedicación que me han prestado durante estos cuatro años.

En segundo lugar, me gustaría agradecer a todos los compañeros del Grupo de Modelización, Simulación Numérica y Matemática Industrial por los medios materiales y humanos de los que he disfrutado durante este tiempo. En especial, agradezco a mis compañeros de doctorado, Andrés Segura, Carlos Chávez e Idulfo Arrocha, porque su amistad ha hecho más ameno este largo viaje. También quisiera agradecer a la Universidad Carlos III de Madrid por financiar la beca doctoral que lo ha hecho posible.

Finalmente, me gustaría agradecer a mis padres y hermanos por su contínuo apoyo, y a Alba, porque con su alegría, comprensión y cariño me ilumina cada día.

Contents

Overview of thesis topics

Radial Basis Function (RBF) methods have become a truly *meshless* alternative for the interpolation of multidimensional scattered data and the solution of partial differential equations (PDEs) on irregular domains. The dependence on the distance between centers makes RBF methods conceptually simple and easy to implement in any dimension or shape of the domain. There are two different formulations for the solution of PDEs: the *global RBF method* and the *local RBF method*.

In the global RBF formulation, the approximate solution is computed in the functional space spanned by a set of translated RBFs. The coordinates of the solution in this space are obtained by collocation. This formulation yields dense differentiation matrices which are spectrally convergent independently of the distribution of RBF centers. The principal drawback is that, as the overall number of centers increases, the condition number of the collocation matrices also increases, what restricts the applicability of the method in practical problems.

To overcome some of the drawbacks of the global RBF method, the local RBF method was independently proposed by several authors which gave the method different names: Shu *et al.* [95] *local multiquadric-based differential quadrature* (LMQDQ) method, Tolstykh *et al.* [100] *RBF in a "finite difference mode"*, Wright [108] *RBF-generated finite differences*. This method can be considered as a natural generalization of classical finite differences (FD). As in FD, the local RBF method approximates a differential operator at a given node as a weighted sum of the values of the sought function at some surrounding nodes. However, while FD methods calculate the weights through polynomial interpolation, the local RBF method does it through an RBF interpolant. Since both, FD and local RBF formulas are identical in form, the local RBF method is commonly denoted as RBF-generated finite differences (RBF-FD) [108].

Unlike the global RBF method, the RBF-FD method lacks spectral accuracy. However, the ability of the method to solve PDEs on irregular domains using highly sparse differentiation matrices, together with the high order

accuracy of the computed solutions, place the method at the forefront of *meshfree* methods.

In this thesis we focus on the RBF-FD method. The state of the art of RBF methods is briefly reviewed in the first chapter. Among the different topics, we introduce some of the concepts that we will use along the thesis and highlight the equivalence between global and local RBF methods, which apparently is not well known in the RBF community. The rest of the thesis can be divided in two parts.

In the first part of the thesis, the convergence properties of the method are addressed. In Chapter 2 and Appendix A, we obtain novel formulas which show the exact dependence of the local truncation error on the inter-nodal distance h, shape parameter ε and stencil size n. These formulas prove the well-known experimental evidence that the accuracy of the method strongly depends on the value of this shape parameter. As $\varepsilon \to 0$, RBFs become flatter and the error decreases until an optimal value which produces the most accurate result is reached. For larger values of the shape parameter, the RBF interpolant converges to a multivariate polynomial and RBF-FD formulas coincide with polynomial based methods. The main result of our study is to analytically derive explicit formulas for the optimal shape parameter that minimizes the local truncation error and produces more accurate results than FD. Contrary to what is commonly believed, such value is independent of the inter-nodal distance to leading order and only depends on the function and its derivatives.

Consequently, the problem of how to select appropriate values for the shape parameter in order to minimize the error of the approximation to a solution of a PDE is of primary concern. Based on the local truncation error formulas derived, we address this problem and propose two algorithms to compute the optimal shape parameters in Chapters 3 and 4. The first of these algorithms is based on finding a *node-independent shape parameter* which minimizes the infinite norm of the global error. We show through numerical experiments that the accuracy of the solution can be improved one or two orders of magnitude with respect to finite differences. We present a second algorithm based on a *node-dependent shape parameter* which minimizes the local truncation error at each node of the domain. To assure the existence of an optimal shape parameter at every node, we make use of the generalized multiquadrics as RBFs. In this way, we are able to obtain significant accuracy improvements with respect to the first algorithm.

In the second part of the thesis, we explore the applicability of the RBF-FD method for the solution of different practical problems. In Chapter 5 we solve some classical elastostatic problems modeled by systems of linear

elliptic PDEs. We apply the algorithms above and show that significant gains in accuracy result from a proper selection of the shape parameter.

The convergence of the approximation can be improved by increasing the size of the stencil. However, exact formulas for the calculation of the optimal shape parameter can only be obtained for the typical FD stencils with a relatively small number of nodes. Going further involves expanding the system of equations which determines the RBF-FD weights, making unattainable the analytical solution.

In this case, a different approach often taken in applications is to adjust the shape parameter in such a way that the condition number of the system matrix is bounded in the region for which the system of equations is still well-conditioned and the error is close to its minimum value. This procedure is analized in Section 1.6 of Chapter 1 and applied in Chapters 6 and 7 to the solution of some non-linear problems in irregular domains. In this way, a three-dimensional combustion model is implemented to high order using large stencil sizes in Chapter 6. The model consists of two convection-diffusion equations coupled through a highly non-linear reaction rate. The satisfactory results obtained motivated us to implement a mathematical model for the study of an idealized micro-rotary engine using RBF-FD, which is the topic of Chapter 7. In this model, it is assumed that the combustible flow field is not affected by the combustion process and can be determined a-priori by solving the steady Navier-Stokes equation. The model of Chapter 6 is then used to simulate the combustion process inside the engine, where the convective term contains the stationary flow field. We present some results which are preliminary but still very encouraging.

Chapter 1

The RBF method

1.1 Outline

Classical methods for the numerical solution of PDEs are based on polynomial interpolation. Local polynomial based methods such as finite differences, finite elements or finite volumes are limited by their algebraic convergence. Global polynomials methods such as spectral methods have exponential convergence, but are limited by being tied to a fixed grid. Over the last decade, Radial Basis Function (RBF) methods have emerged as a truly meshless alternative for the interpolation of multidimensional scattered data and the solution of PDEs on irregular domains. Its dependence on the distance between nodes makes RBF methods conceptually simple and easy to implement in any dimension and/or shape of the domain. This fact and the possibility of obtaining high order approximations independently of the node distribution represent the main advantages of RBF methods.

In the present chapter we overview the state of the art of RBF methods. We introduce the RBF interpolation and establish its unisolvency conditions in Section 1.2. Global and local RBF methods for the solution of PDEs are respectively introduced in Section 1.3 and 1.4. The equivalence between both methods is pointed out in Section 1.5. This equivalence allows us to formulate them together as a single method. Finally, Section 1.6 is devoted to explain the dependence of RBF methods on the shape parameter and to discuss some of the usual strategies to select it which have been proposed in the past.

1.2 Scattered data interpolation using RBFs

Interpolation of scattered data is a common issue in many engineering and scientific problems. Given a set of N data points $\{\mathbf{x}_j\}_{j=1}^N$, $\mathbf{x}_j \in \mathbb{R}^d$, and the corresponding data values $\{f_j\}_{j=1}^N$, the usual procedure is to look for an interpolant $s(\mathbf{x})$ as a linear combination of certain basis functions $\{\psi_j(\mathbf{x})\}_{j=1}^N$ which span the functional space (such as polynomials, trigonometric functions, etc),

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \psi_j(\mathbf{x}), \tag{1.1}$$

and determine the corresponding expansion coefficients $\{\lambda_j\}_{j=1}^N$ by solving the linear system

$$\begin{bmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \dots & \psi_N(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \dots & \psi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \dots & \psi_N(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \tag{1.2}$$

which results from the interpolation conditions $s(\mathbf{x}_j) = f_j$, $j = 1, \dots, N$. In one-dimensional problems, this linear system is guaranteed to be non-singular whenever the data points are distinct. However, in more than 1D this is no longer assured due to the Mairhuber-Curtis theorem [18, 78]. The point is that in more than one dimension it is possible to move the nodes continuously so that two nodes end up interchanged without coinciding at any time. As a result, two rows are interchanged in the matrix of equation (1.2) and its determinant changes sign. Therefore, the determinant must have been zero somewhere along the way, resulting in a singular system. Since there are an infinite number of node configurations that will yield a singular interpolation problem, the procedure above is limited to regular grids, for which handling irregular geometries or carrying out local node refinement is almost unattainable [39].

To bypass this problem, R. L. Hardy was the first who used *radial basis functions* (RBF) as basis functions in the expansion (1.1) to solve a cartography problem in 1968 [50]. RBFs are a set of functions centered at a data point $\mathbf{x}_j$, called *RBF center*, radially symmetric about it. There are two types of RBFs: the *infinitely smooth* and the *piecewise smooth* RBFs. All infinitely smooth RBFs depend on a shape parameter ε which controls the shape of the functions (at this point assume that is some fixed non-zero real value). RBFs

Table 1.1: Some commonly used radial basis functions

Infinitely smooth RBFs	
Multiquadric (MQ)	$\sqrt{1+\varepsilon^2 r^2}$
Inverse Multiquadric (IMQ)	$1/\sqrt{1+\varepsilon^2 r^2}$
Inverse quadratic (IQ)	$1/\left(1+\varepsilon^2 r^2\right)$
Gaussian (GA)	$e^{-\varepsilon^2 r^2}$
Piecewise smooth RBFs	
Piecewise polynomial (R_n)	r^n, n odd
Thin Plate Spline (TPS_n)	$r^n \ln r$, n even
Compact support (Wendland)	$(1-r)_+^n p(r)$, p certain polynomials, $n \in \mathbb{N}$

from this group are $C^\infty(0,\infty)$ and can provide spectral accuracy [76, 77, 90]. On the other hand, piecewise smooth RBFs leads to algebraic convergence [8, 27, 106]. Many examples of commonly used RBFs are shown in Table 1.1, where the variable $r = \|\mathbf{x}-\mathbf{x}_j\|$ refers to the euclidean norm. Hardy used multiquadrics (MQ) as RBF

$$\phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right) = \sqrt{c^2+\|\mathbf{x}-\mathbf{x}_j\|^2}, \tag{1.3}$$

which belongs to the group of infinitely smooth RBFs. Notice that this definition is equivalent to the one given in Table 1.1 changing $c = 1/\varepsilon$ and ignoring the scale factor ε. Many authors prefer the definition of Table 1.1 since all the RBFs become flatter as the shape parameter tends to zero.

Using RBFs, the expansion (1.1) transforms into

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right). \tag{1.4}$$

There are two ways for determining the expansion coefficients $\{\lambda_j\}_{j=1}^N$, collocation and least squares. In collocation, the RBF centers coincide with the data locations so that $s(\mathbf{x}_j) = f_j$, $j = 1,\ldots,N$ is enforced at the RBF centers. In least squares, the number of RBF centers N is less than the number of data locations M and the system is solved in a least-squares sense. The procedure we adopt in this thesis is collocation, for which the expansion coefficients are given by the linear system of equations

$$A\bar{\lambda} = \bar{f}, \tag{1.5}$$

where the entries of A-matrix are

$$A_{ij} \quad = \quad \phi(||\mathbf{x}_i - \mathbf{x}_j||), \qquad i,j = 1, \ldots, N,$$

and the vectors $\bar{\lambda}$ and $\bar{f}$ are $\bar{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$ and $\bar{f} = [f_1, \ldots, f_N]^T$.

In this case, moving two nodes continuously so that they end up interchanged results in two rows and two columns interchanged in the interpolation matrix A. The sign of the determinant is unaffected and therefore, it does not imply a singularity in the system of equations. In fact, it is well known that under some conditions for the RBF $\phi(r)$, A-matrix is non-singular no matter how the nodes are scattered in any number of dimensions. The first sufficient conditions were given by Schoenberg [92] in 1938. From this result, the non-singularity is guaranteed if the RBF is such that it gives rise to a *positive definite* matrix, i.e. a symmetric real-valued matrix for which any of the following equivalent conditions hold:

1. All eigenvalues λ_i of A are positive.
2. $\bar{c}^T A \bar{c} > 0$, for $\bar{c} \neq 0$.
3. All leading principal minors of A are positive.

Such as RBF $\phi(r)$ is said to be *positive definite* since it verifies

$$\bar{c}^T A \bar{c} = \sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j \phi\left(\|\mathbf{x}_i - \mathbf{x}_j\|\right) > 0.$$

Some common RBFs (like MQ or linear RBFs, $\phi(r) = r$) fail to be positive definite.

Hardy's MQ interpolation method went unnoticed until 1979, when the mathematician Richard Franke compared various methods to solve the scattered interpolation problem [45, 46]. He concluded that Hardy's MQ interpolation method was the best, followed by Duchon's thin plate spline (TPS) [24]. Additionally, he conjectured that the collocation matrix of the method was invertible and that the method was well-posed. It was Michelli in 1986 who proved this conjecture and extended Schoenberg's result so that a larger class of functions could be considered [79]. Although MQ or linear RBFs do not yield positive definite A-matrices, he proved that they are non-singular with one positive and $n-1$ negative eigenvalues. Moreover, he also proved the non-singularity of the interpolation matrix for a wider class of RBFs (*strictly conditionally positive definite* functions) through the addition of polynomial terms in the RBF interpolant.

To establish this result, consider the augmented RBF interpolant, which takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \, \phi(||\mathbf{x} - \mathbf{x}_j||) + \sum_{i=1}^{M} \beta_i \, p_i(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^d, \tag{1.6}$$

where $\phi(r)$ is some radial function, $||\cdot||$ is the euclidian norm and $\{p_k(\mathbf{x})\}_{k=1}^{M}$ form a basis of the space Π_{m-1}^d of all polynomials up to degree $m-1$ in $\mathbb{R}^d$. The number of elements of the basis is given by $M = \binom{m-1+d}{m-1}$.

The expansion coefficients $\{\lambda_j\}_{j=1}^{N}$ and $\{\beta_i\}_{i=1}^{M}$ are determined by enforcing the interpolation conditions

$$s(\mathbf{x}_j) = f_j, \qquad j = 1, \ldots, N, \tag{1.7}$$

and the orthogonality conditions

$$\sum_{j=1}^{N} \lambda_j p_k(\mathbf{x}_j) = 0, \qquad k = 1, \ldots, M. \tag{1.8}$$

It results in the block linear system of equations

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \bar{\lambda} \\ \bar{\beta} \end{bmatrix} = \begin{bmatrix} \bar{f} \\ \bar{0} \end{bmatrix}, \tag{1.9}$$

where A is the interpolation matrix from equation (1.5), P is the $N \times M$ matrix with entries $p_k(\mathbf{x}_j)$ for $j = 1, \ldots, N$ and $k = 1, \ldots, M$, O is an $M \times M$ zero matrix, $\bar{0}$ is a zero vector of length M, $\bar{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$, $\bar{\beta} = [\beta_1, \ldots, \beta_M]^T$ and $\bar{f} = [f_1, \ldots, f_N]^T$.

The theorem which guarantees the solvability of the system of equations (1.9) requires the definition of a *strictly conditionally positive definite* function [27].

Definition 1. *A real-valued continuous even function ϕ is called* strictly conditionally positive definite of order m *on $\mathbb{R}^d$ if*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j \phi \left(\|\mathbf{x}_i - \mathbf{x}_j\| \right) > 0 \tag{1.10}$$

for any N pairwise distinct points $\{\mathbf{x}_j\}_{j=1}^{N}$, $\mathbf{x}_j \in \mathbb{R}^d$, and $\bar{\lambda} = [\lambda_1, \ldots, \lambda_N]^T \in \mathbb{R}^N \setminus \{\bar{0}\}$ satisfying

$$\sum_{j=1}^{N} \lambda_j p(\mathbf{x}_j) = 0,$$

for any real-valued polynomial p of degree at most $m-1$.

From the above definition, A-matrix in (1.9) can be interpreted as a positive definite matrix on the space of vectors $\bar{\lambda}$ orthogonal to d-variate polynomials up to degree $m-1$. The next theorem guarantees the solvability of the system of equations [27]:

Theorem 1. *If the real-valued even function ϕ is strictly conditionally positive definite of order m on $\mathbb{R}^d$ and the matrix P in (1.9), with entries*

$$P_{jk} = p_k(\mathbf{x}_j) \qquad j = 1, \ldots, N,\ k = 1, \ldots, M,$$

has full column-rank, then the system of linear equations (1.9) is uniquely solvable.

To illustrate the scope of this theorem, consider Hardy's interpolant (1.4) for some RBF. The system of equations which determines the interpolation coefficients $\{\lambda_j\}_{j=1}^N$ is given by (1.5). If the corresponding A-matrix is not positive (or negative) definite, there may not be guarantee to be non-singular. However, including some low order polynomial terms in the RBF interpolant, extents the matrix with some additional rows and columns and makes it positive (or negative) in the constrained parameter spaces. Theorem 1 guarantees now the non-singularity of the system of equations. For example, including a constant will create negative definite interpolation matrices for the MQ and linear RBFs. Including also linear terms will similarly create positive definite interpolation matrices in the TPS and cubic RBFs $\phi(r) = r^3$ [39, 79].

In this thesis we will use RBFs which do not require polynomial augmentation to assure the non-singularity of A-matrix (such as GA or MQ). In this way, we simplify the algorithm and reduce the computational cost.

1.3 Global RBFs for solving PDEs

Kansa was the first who proposed the use of RBF interpolants for solving PDEs in 1990 [55, 56]. In these articles, he solved problems from fluid mechanics approximating the spatial derivatives with the multiquadric interpolant (1.4). The idea is to consider that the solution of the PDE is approximated by the RBF interpolant centered at $\{\mathbf{x}_j\}_{j=1}^N$. In this way, if $\mathcal{L}$ is the spatial differential operator, $\mathcal{L}u$ can be approximated as

$$\mathcal{L}u\left(\mathbf{x}\right) \approx \mathcal{L}s\left(\mathbf{x}\right) = \sum_{j=1}^{N} \lambda_j \mathcal{L}\phi\left(\left\|\mathbf{x} - \mathbf{x}_j\right\|\right). \tag{1.11}$$

To determine the expansion coefficients $\{\lambda_j\}_{j=1}^N$ for a time dependent problem, collocation of the interpolation conditions (1.7) at time t_k yields the

expansion coefficients

$$\bar{\lambda} = A^{-1}\,\bar{u}^{(k)}, \tag{1.12}$$

which is guaranteed to be non-singular from the theory of Section 1.2. Substituting (1.12) into (1.11) allows us to approximate $\mathcal{L}u$ at time t_k not only at the collocation centers but at any point $\mathbf{x}$ in the domain.

For steady PDEs,

$$\left\{ \begin{array}{rcll} \mathcal{L}u(\mathbf{x}) & = & f(\mathbf{x}) & \text{in } \Omega, \\ \mathcal{B}u(\mathbf{x}) & = & g(\mathbf{x}) & \text{on } \partial\Omega, \end{array} \right. \tag{1.13}$$

collocation at the N_B boundary nodes and the N_I interior nodes ($N = N_I + N_B$) leads to the block linear system

$$\left[\begin{array}{c} \bar{g} \\ \bar{f} \end{array} \right] = \left[\begin{array}{c} A_{\mathcal{B}} \\ A_{\mathcal{L}} \end{array} \right] \left[\begin{array}{c} \\ \bar{\lambda} \\ \\ \end{array} \right] \tag{1.14}$$

which determines $\{\lambda_j\}_{j=1}^{N}$, where

$$\begin{array}{rcll} (A_{\mathcal{B}})_{ij} & = & \mathcal{B}\phi(||\mathbf{x}-\mathbf{x}_j||)|_{\mathbf{x}=\mathbf{x}_i}\,, & i = 1,\ldots,N_B, \\ (A_{\mathcal{L}})_{ij} & = & \mathcal{L}\phi(||\mathbf{x}-\mathbf{x}_j||)|_{\mathbf{x}=\mathbf{x}_i}\,, & i = N_B+1,\ldots,N, \end{array}$$

for $j = 1,\ldots,N$. Notice that the collocation matrix above is not symmetric. This is the reason why Kansa's method is known as *asymmetric collocation method*. Although it is extremely rare in practice, it may exist grid configurations for which the rows of the asymmetric matrix (1.14) are not linearly independent and becomes singular [52].

A variation that leads to a *symmetric collocation method* which is guaranteed to be non-singular for the appropriate choice of ϕ was derived by Wu [110] and Fasshauer [25]. The idea is to modify the basis functions in the interpolant by applying the linear operators $\mathcal{L}$ and $\mathcal{B}$ to each basis function centered at the corresponding node point. Therefore, the basis function is $\mathcal{B}\phi$ for boundary points and $\mathcal{L}\phi$ for interior points such that

$$s(\mathbf{x}) = \sum_{j=1}^{N_B} \lambda_j\, \mathcal{B}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\boldsymbol{\xi}=\mathbf{x}_j} + \sum_{j=N_B+1}^{N} \lambda_j\, \mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\boldsymbol{\xi}=\mathbf{x}_j}\,. \tag{1.15}$$

By collocation, it yields the block linear system of equations

$$\left[\begin{array}{c} \bar{g} \\ \bar{f} \end{array} \right] = \left[\begin{array}{cc} A_{\mathcal{B}\mathcal{B}_{\boldsymbol{\xi}}} & A_{\mathcal{B}\mathcal{L}_{\boldsymbol{\xi}}} \\ A_{\mathcal{L}\mathcal{B}_{\boldsymbol{\xi}}} & A_{\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}} \end{array} \right] \left[\begin{array}{c} \\ \bar{\lambda} \\ \\ \end{array} \right], \tag{1.16}$$

where

$$\begin{aligned}\left(A_{\mathcal{B}\mathcal{B}_{\boldsymbol{\xi}}}\right)_{ij} &= \mathcal{B}\mathcal{B}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_j}, \qquad j=1,\ldots,N_B,\\ \left(A_{\mathcal{B}\mathcal{L}_{\boldsymbol{\xi}}}\right)_{ij} &= \mathcal{B}\mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_j}, \qquad j=N_B+1,\ldots,N,\end{aligned}$$

for $i = 1, \ldots, N_B$, and

$$\begin{aligned}\left(A_{\mathcal{L}\mathcal{B}_{\boldsymbol{\xi}}}\right)_{ij} &= \mathcal{L}\mathcal{B}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_j}, \qquad j=1,\ldots,N_B,\\ \left(A_{\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}}\right)_{ij} &= \mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_j}, \qquad j=N_B+1,\ldots,N,\end{aligned}$$

for $i = N_B + 1, \ldots, N$. Note that $A_{\mathcal{B}\mathcal{L}_{\boldsymbol{\xi}}} = \left(A_{\mathcal{L}\mathcal{B}_{\boldsymbol{\xi}}}\right)^T$, so that the collocation matrix is symmetric.

As in the time-dependent case, substituting $\{\lambda_j\}_{j=1}^N$ into the corresponding RBF interpolant allows us to approximate the solution u at any point $\mathbf{x}$ in the domain.

1.4 Local RBFs for solving PDEs

Global RBF methods yields full differentiation matrices which requires $O(N^3)$ floating point operations to solve the linear system of equations (1.12), (1.14) or (1.16) and a memory cost $O(N^2)$. Therefore, the number of nodes are limited and so its applicability. Several different approaches have been proposed to deal with this issue such as fast multipole methods [3, 4, 13], domain decomposition [2, 70, 72, 112] and compactly supported RBFs [8, 26, 37].

A different approach was first introduced by Tolstykh in 2000 in a conference presentation [99] and published later in 2003 by Tolstykh [100] and Shu [95]. As in classical finite differences (FD), the idea is to approximate $\mathcal{L}u$ at $\mathbf{x}_i$ by a linear combination of function $u(\mathbf{x})$ evaluated at the n-node stencil centered on $\mathbf{x}_i$,

$$\mathcal{L}u|_{\mathbf{x}_i} \approx \sum_{j\in\sigma_i} \alpha_{ij} u(\mathbf{x}_j), \tag{1.17}$$

where α_{ij} are the weighting coefficients and $\sigma_i = \{\sigma_i(k)\}_{k=1}^n$ is a set which contains the indices of the nodes which form the stencil $\{\mathbf{x}_{\sigma_i(k)}\}_{k=1}^n$. Instead of using polynomial interpolants as in FD, the weighting coefficients are obtained approximating u in (1.17) by the RBF interpolant (1.4) evaluated at the stencil $\{\mathbf{x}_{\sigma_i(k)}\}_{k=1}^n$, which leads to the system of equations

$$A\bar{\alpha}_i \;=\; \mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i}, \tag{1.18}$$

where

$$A_{kj} \;=\; \phi(||\mathbf{x}_{\sigma_i(k)} - \mathbf{x}_{\sigma_i(j)}||), \qquad k = 1, \ldots, n, \;\; j = 1, \ldots, n, \quad (1.19)$$

$$\bar{\alpha}_i = \left[\alpha_{i\sigma_i(1)}, \; \ldots, \; \alpha_{i\sigma_i(n)}\right]^T. \quad (1.20)$$

and

$$\mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i} = \left[\mathcal{L}\phi(||\mathbf{x} - \mathbf{x}_{\sigma_i(1)}||)\big|_{\mathbf{x}=\mathbf{x}_i}, \; \ldots, \; \mathcal{L}\phi(||\mathbf{x} - \mathbf{x}_{\sigma_i(n)}||)\big|_{\mathbf{x}=\mathbf{x}_i}\right]^T. \quad (1.21)$$

The collocation matrix (1.19) is equivalent in form to the one obtained in the interpolation case from Section 1.2, which is guaranteed to be non-singular for any node configuration whenever the RBF chosen is positive definite.

Thereby, the local RBF method, known as *RBF-generated FD* (RBF-FD), is considered a natural generalization of classical finite differences [44, 109]. It has been successfully applied to a wide range of problems including elliptic problems [100, 109], Navier-Stokes equations [14, 87, 95], convective-diffusion [11] and pure convective PDEs [38, 40, 67].

As in FD, the order of convergence of the approximation depends on the number of nodes per stencil. An alternative way to increase the order of the RBF-FD approximations keeping constant the stencil size, known as RBF-HFD, was first proposed by Wright in [109]. Following the idea of compact FD formulas [17, 69], $\mathcal{L}u$ at $\mathbf{x}_i$ is approximated by a linear combination of function $u(\mathbf{x})$ and $\mathcal{L}u(\mathbf{x})$ evaluated at the n-node stencil centered on $\mathbf{x}_i$,

$$\mathcal{L}u\big|_{\mathbf{x}_i} \approx \sum_{j \in \sigma_i} \alpha_{ij} u(\mathbf{x}_j) + \sum_{j \in \hat{\sigma}_i} \omega_{ij} \, \mathcal{L}u\big|_{\mathbf{x}_j}, \quad (1.22)$$

where α_{ij} and ω_{ij} are the weighting coefficients, $\sigma_i = \{\sigma_i(k)\}_{k=1}^n$ is the set which contains the indices of the nodes which form the stencil $\{\mathbf{x}_{\sigma_i(k)}\}_{k=1}^n$ and $\{\hat{\sigma}_i(k)\}_{k=1}^m \subseteq \{\sigma_i(k)\}_{k=1}^n$ is the set which contains the indices of the nodes which expands (1.22) compactly.

To determine the weighting coefficients, function $u(\mathbf{x})$ in (1.22) is interpolated by an Hermite RBF interpolant,

$$s(\mathbf{x}) = \sum_{k \in \sigma_i} \lambda_k \, \phi(||\mathbf{x} - \mathbf{x}_k||) + \sum_{k \in \hat{\sigma}_i} \lambda_k \, \mathcal{L}_{\boldsymbol{\xi}} \phi(||\mathbf{x} - \boldsymbol{\xi}||)\big|_{\boldsymbol{\xi}=\mathbf{x}_k}, \quad (1.23)$$

which leads to the block linear system of equations

$$\begin{bmatrix} A & A_{\mathcal{L}_\xi} \\ A_{\mathcal{L}} & A_{\mathcal{L}\mathcal{L}_\xi} \end{bmatrix} \begin{bmatrix} \bar{\alpha}_i \\ \bar{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathcal{L}\bar{\phi}_i \\ \mathcal{L}\mathcal{L}_\xi \bar{\phi}_i \end{bmatrix}, \quad (1.24)$$

where matrix A and vectors $\bar{\alpha}_i$ and $\mathcal{L}\bar{\phi}_i$ are respectively given by (1.19), (1.20) and (1.21), vectors $\bar{\omega}_i$ and $\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\bar{\phi}_i$ are

$$\bar{\omega}_i = \left[\omega_{i\sigma_i(1)}, \ldots, \omega_{i\sigma_i(m)}\right]^T,$$

$$\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\bar{\phi}\big|_{\mathbf{x}_i} = \left[\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)\big|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_{\hat{\sigma}_i(1)}}, \ldots, \mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}-\boldsymbol{\xi}||)\big|_{\mathbf{x}=\mathbf{x}_i,\,\boldsymbol{\xi}=\mathbf{x}_{\hat{\sigma}_i(m)}}\right]^T,$$

and the block matrices are

$$\begin{aligned}
\left(A_{\mathcal{L}}\right)_{jk} &= \mathcal{L}\phi(||\mathbf{x}-\mathbf{x}_{\sigma_i(k)}||)\big|_{\mathbf{x}=\mathbf{x}_{\sigma_i(j)}}, & j=1,\ldots,m, \quad k=1,\ldots,n,\\
\left(A_{\mathcal{L}_{\boldsymbol{\xi}}}\right)_{jk} &= \mathcal{L}_{\boldsymbol{\xi}}\phi(||\mathbf{x}_{\sigma_i(j)}-\boldsymbol{\xi}||)\big|_{\boldsymbol{\xi}=\mathbf{x}_{\sigma_i(k)}}, & j=1,\ldots,n, \quad k=1,\ldots,m,\\
\left(A_{\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}}\right)_{jk} &= \mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\phi(||\boldsymbol{\xi}-\mathbf{x}||)\big|_{\mathbf{x}=\mathbf{x}_{\sigma_i(j)},\,\boldsymbol{\xi}=\mathbf{x}_{\sigma_i(k)}}, & j=1,\ldots,m, \quad k=1,\ldots,m.
\end{aligned}$$

Notice that the system of equations (1.24) is symmetric since $A_{\mathcal{L}} = \left(A_{\mathcal{L}_{\boldsymbol{\xi}}}\right)^T$.

Although for the RBFs considered it is not necessary to include some low order polynomial terms in the RBF interpolant (1.6) for the solvency of the system of equations (1.18) or (1.24), it has the advantage of improving the accuracy of the derivative approximations, making it exact for all polynomials of the same order [109]. In this case, the RBF-FD weighting coefficients are given by

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \bar{\alpha}_i \\ \bar{\mu}_i \end{bmatrix} = \begin{bmatrix} \mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i} \\ \mathcal{L}\bar{p}_i \end{bmatrix}, \tag{1.25}$$

and the RBF-HFD weighting coefficients by

$$\begin{bmatrix} A & A_{\mathcal{L}_{\boldsymbol{\xi}}} & P \\ A_{\mathcal{L}} & A_{\mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}} & P_{\mathcal{L}} \\ P^T & P_{\mathcal{L}}^T & O \end{bmatrix} \begin{bmatrix} \bar{\alpha}_i \\ \bar{\omega}_i \\ \bar{\mu}_i \end{bmatrix} = \begin{bmatrix} \mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i} \\ \mathcal{L}\mathcal{L}_{\boldsymbol{\xi}}\bar{\phi}\big|_{\mathbf{x}_i} \\ \mathcal{L}\bar{p}_i \end{bmatrix}, \tag{1.26}$$

where P is given in equation (1.9), $\bar{p}_i = \left[p_1(\mathbf{x}_i), \ldots, p_M(\mathbf{x}_i)\right]^T$ and $\bar{\mu}_i$ is a dummy vector related with the vector of coefficients $\bar{\beta}$ in the RBF interpolant (1.6).

1.5 Equivalence of global and local formulations

Consider the global RBF approximation to $\mathcal{L}u(\mathbf{x})$ from equation (1.11),

$$\mathcal{L}u(\mathbf{x}) \approx \mathcal{L}s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \mathcal{L}\phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right).$$

Substituting the expansion coefficients $\{\lambda_j\}_{j=1}^N$ given by (1.12) yields

$$\mathcal{L}u\,(\mathbf{x}) \approx \mathcal{L}\bar{\phi}\,(\mathbf{x})\,A^{-1}\,\bar{u}, \tag{1.27}$$

where the row vector $\mathcal{L}\bar{\phi}\,(\mathbf{x})$ is given by

$$\mathcal{L}\bar{\phi}\,(\mathbf{x}) = \left[\mathcal{L}\phi\left(\|\mathbf{x}-\mathbf{x}_1\|\right),\,\ldots,\,\mathcal{L}\phi\left(\|\mathbf{x}-\mathbf{x}_N\|\right)\right].$$

From this equation, it is clear that the global RBF method approximates $\mathcal{L}u$ at $\mathbf{x}_i$ as

$$\mathcal{L}u\big|_{\mathbf{x}_i} \approx \mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i}\,A^{-1}\bar{u}, \tag{1.28}$$

or equivalently, as

$$\mathcal{L}u\big|_{\mathbf{x}_i} \approx \sum_{j=1}^{N}\alpha_{ij}u(\mathbf{x}_j), \tag{1.29}$$

where the weighting coefficients α_{ij} are the unknowns of the system of equations $\bar{\alpha}_i\,A = \mathcal{L}\bar{\phi}\big|_{\mathbf{x}_i}$. Notice that this is equivalent to the transpose of the linear system of equations (1.18), which determines the RBF-FD weights, but with all the nodes from the domain included in the stencil.

Therefore, both global and local approximations are constructed equivalently. The only difference resides in the number of supporting nodes per stencil. While global RBF methods uses all the nodes N in the domain as stencil, the local RBF method uses stencils formed by $n < N$ nodes, usually the $n-1$ closest to the stencil center, yielding sparse differentiation matrices. This result is general. It can be shown that global RBF approximation using Hermite RBF interpolants (1.23) leads to the system of equations (1.24) which determines the RBF-HFD weights.

In view of this equivalence, one advantage of the global RBF method which is overlooked with the local RBF formulation of Section 1.4 is the possibility of evaluate $\mathcal{L}u$ at a point $\mathbf{x}$ different to the stencil nodes. To emphasize this equivalence and to exploit all the capabilities of both methods, we formulate both together as a single method.

Given a spatial differential operator $\mathcal{L}$ and a set of RBF centers $\{\mathbf{x}_j\}_{j=1}^N \in \Omega$, $\mathcal{L}u$ is approximated at $\mathbf{x}$ by a linear combination of function u evaluated at the stencil nodes $\{\mathbf{x}_{\sigma(j)}\}_{j=1}^n \subseteq \{\mathbf{x}_j\}_{j=1}^N$, usually the closest, such that

$$\mathcal{L}u(\mathbf{x}) \approx \sum_{j\in\sigma}\alpha_j(\mathbf{x})u(\mathbf{x}_j), \tag{1.30}$$

where the weighting coefficients are given by

$$\bar{\alpha}(\mathbf{x}) = \mathcal{L}\bar{\phi}(\mathbf{x})A^{-1}. \tag{1.31}$$

In this equation, the row vector $\mathcal{L}\bar{\phi}(\mathbf{x})$ is equal to

$$\mathcal{L}\bar{\phi}(\mathbf{x}) = \left[\mathcal{L}\phi(||\mathbf{x} - \mathbf{x}_{\sigma(1)}||), \ldots, \mathcal{L}\phi(||\mathbf{x} - \mathbf{x}_{\sigma(n)}||)\right], \tag{1.32}$$

and A-matrix is given by

$$A_{ij} \quad = \quad \phi(||\mathbf{x}_{\sigma(i)} - \mathbf{x}_{\sigma(j)}||), \tag{1.33}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.

Notice that the evaluation node $\mathbf{x}$ can now be different not only to any of the nodes which form the stencil $\{\mathbf{x}_{\sigma(j)}\}_{j=1}^{n}$ but also to any of the nodes $\{\mathbf{x}_j\}_{j=1}^{N}$ which discretizes the domain. This fact makes possible to perform node refinement of the domain when solving PDE problems with both global or local RBF methods or to interpolate a function locally, considering only the *nth*-closest nodes to $\mathbf{x}$.

1.6 Shape parameter and ill-conditioning

The accuracy and the stability of RBF methods depend on the number of nodes N and the value of the shape parameter ε. It is well-known that for the global method the rate of convergence is spectral as either ε or h go to zero [76, 77, 90], where h is the fill distance of the node set defined as the radius of the largest possible empty ball that can be placed among the N nodes in the domain.

The convergence can be discussed in terms of two different types of approximation: *stationary*, where the number of nodes N is fixed and the shape parameter ε is refined towards zero, and *non-stationary*, where the value of the shape parameter is fixed and N is increased. However, both procedures make the condition number of A-matrix to grow exponentially with the consequent effect on the stability of the method [80, 90]. The reason is that as $\varepsilon h \to 0$, the RBFs becomes flatter and flatter. As a consequence, the basis functions become more linearly dependent and the condition number of the A-matrix grows, together with the accuracy. This trade-off between accuracy and ill-conditioning is known as "Schaback's uncertainty principle" [90]. Figure 1.1 and Figure 1.2 show an example of the dependence of the error and the condition number on the shape parameter for a fixed N (stationary approximation) and the dependence of the error and the condition number

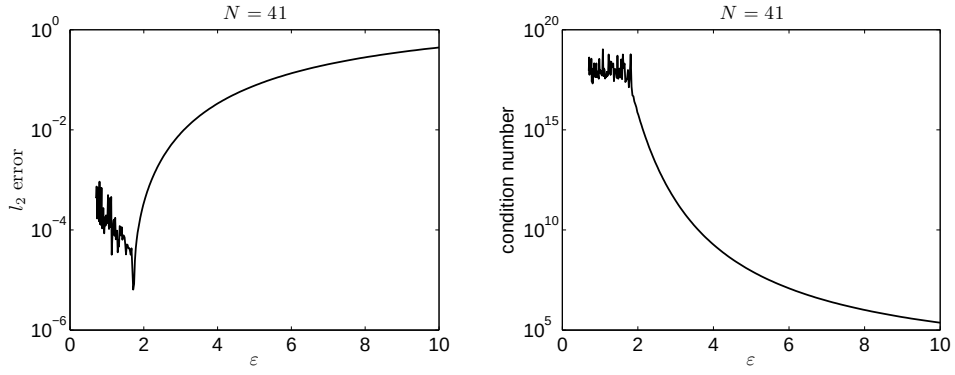


Figure 1.1: The l_2 error (left) and the condition number (right) against the shape parameter ε when approximating the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$ using global MQ RBF with $N = 41$.

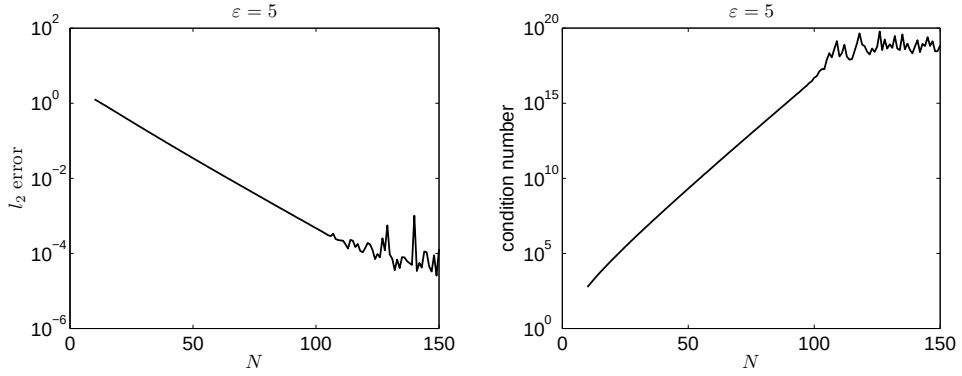


Figure 1.2: The l_2 error (left) and the condition number (right) against the number of nodes N when approximating the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$ using global MQ RBF with $\varepsilon = 5$.

on N for a fixed ε (non-stationary approximation) when applying the global MQ RBF method, respectively.

The problem of how to select appropriate values for the shape parameter in interpolation and in the solution of PDEs has been of primary concern both from the theoretical and the application points of view. Some of the proposed techniques address the problem of how to select a single (constant) value of the shape parameter, while others address the problem of selecting different shape parameters values for each node.

For MQ interpolation problems in 2D, Hardy [50] suggested the use of $c = 0.815\,d$, where d is the mean distance from each point to its nearest neighbor ($d = h$ for equispaced nodes). On the other hand, Franke [46] recommended $c = 1.25D/\sqrt{N}$, where D is the diameter of the smallest circle containing all data points ($c = 1.25\sqrt{2}h$ for equispaced nodes). Notice that both strategies are similar since $d \propto \frac{D}{\sqrt{N}}$. A different approach was taken by Carlson and Foley [9], who pointed out that the optimal value of c was strongly dependent on the interpolated function and essentially independent on the number and location of the interpolation nodes. They also presented an algorithm that yields an effective value of c. Rippa [83] proposed a *leave-one-out* algorithm to estimate the interpolation error and use it to compute an optimal value of the shape parameter. The former approach has also been adapted for the solution of PDEs with the global RBF method by Fasshauer and Zhang [29] and applied by Ferreira *et al* [29, 33, 34]. Regarding spatially varying shape parameters, Kansa and Carlson [57] showed through numerical experiments that using a node dependent value of c for interpolation gives better accuracy than a single (constant) one. The optimal value was determined by numerically minimizing the root mean square error. Kansa and Hon [58] explored the benefits of using spatially varying shape parameters with the global RBF method. They suggested that a variable shape parameter should be related to the local curvature of the solution, and proposed an experimentally based formula to compute its value. Driscoll and Heryudono [23] proposed an adaptive algorithm based on computing residuals on a finer grid and using this information to remove or add nodes. The shape parameters were also adaptively varied by taking them proportional to the local internode distance. Wertz *et al.* [107] used numerical experiments to show that the shape parameter should be significantly higher at boundary nodes than at interior nodes.

Many studies [12, 53, 54] have shown that the optimal value of the shape parameter frequently lies in a region where the interpolant is ill-conditioned and optimal accuracy cannot be obtained. As a result, a significant amount of work has been carried out to address the behavior of the interpolant in the

limit $\varepsilon \to 0$. Driscoll and Fornberg showed that the limiting interpolant in 1D is the Lagrange interpolating polynomial under some conditions on the basis function. Fornberg *et al* [44, 65] later showed that if a limiting interpolant exists, it must be a multivariate polynomial. Moreover, they showed that the divergence of the interpolant occurs only in some special cases for which the polynomial interpolant was not unique for the given data set. An exception is the gaussian RBF, which was proved to never diverge by Schaback [91].

The existence of a limiting interpolant shows that ill-conditioning is not intrinsic to RBF methods, but it is due to a numerically ill-conditioned algorithm. In this context, some novel algorithms haven been proposed to circumvent ill-conditioning such as the *Contour-Padeé* algorithm [43], the *RBF-QR* algorithm [28, 40, 42] and more recently, the *RBF-GA* algorithm [41]. The first is based on allowing ε to be complex and evaluate the interpolant by a contour integral. The last two algorithms are based upon changing the basis to one that is more suited for numerics, while staying in the same approximation space. The main drawback of these algorithms is their computational cost, which is lower than using high precision arithmetic [41], but still higher than directly solving the system of equations (around 10 times higher for the quickest algorithm RBF-GA).

In the global RBF method, an alternative approach often taken in applications is to allow both N and ε to vary. As N increases, the shape parameter is adjusted so that the condition number of the system matrix is bounded in the range $\kappa_{min} \leq \kappa \leq \kappa_{max}$. If κ_{min} and κ_{max} are chosen properly, this strategy yields a value of the shape parameter in the region for which the system of equations is well-conditioned and the error is close to its minimum value. Figure 1.3 shows the results of applying this strategy to the approximation of the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$. Typically, the error decreases until a value of N is reached, for which the convergence ceases and a saturation error remains [27, 89]. On the other hand, the shape parameter increases at a simple rate with N to maintain the desired conditioning.

In the local RBF method, a similar strategy is usually followed [38, 88]. Given a set of N nodes, the differential operator is approximated at some point using stencils of n nodes. As n increases, the shape parameter is also adjusted so that the condition number of the system matrix is bounded in the range desired. Figure 1.4 shows the dependence of the l_2 error versus the stencil size n in the approximation of the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$, where $N = 101$ and $10^{14} \leq \kappa \leq 10^{15}$. Notice that in this case, the minimum error is reached for $n \approx 10$. From there on, the convergence ceases and a saturation error remains. Notice also from Figures 1.3 and 1.4 the equivalency of global and local RBF methods, for which the same error

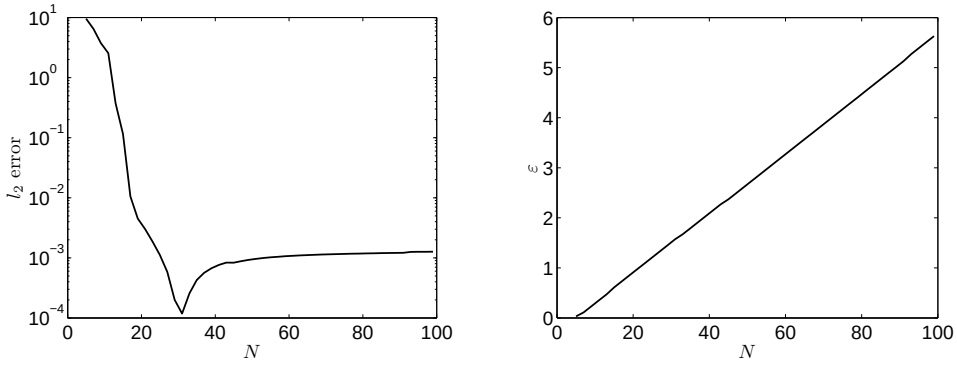


Figure 1.3: The l_2 error (left) and the shape parameter ε (right) against the number of nodes N using a global MQ RBF approximation for the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$ keeping the condition number of A-matrix bounded in the range $10^{14} \leq \kappa \leq 10^{15}$.

is obtained when $n = N$.

Besides the l_2 error, it is interesting to calculate the local approximation error at different x values. In this way, we analyze the difference between the error committed at boundary nodes and interior nodes. Table 1.2 shows the error in the approximation of the first derivative of $f = e^{\sin(\pi x)}$ at some x-values for different stencil sizes n. Notice that the errors are much higher at boundary nodes, for which the error dependence with n is similar to the l_2 error shown in Figure 1.4. At interior nodes, the approximation error yields for $n \approx 15 - 20$ comparable results to those obtained for $n = N$, when global and local methods are equivalent. Although global RBF methods converge at an exponential rate as the shape parameter and the inter-nodal distance between centers are refined [77], the condition number of the system matrix also grows and the convergence ceases at a certain point. As a result, the local RBF method is usually able to match the accuracy of the global method with some relatively small stencil size. In two and three dimensions, stencil sizes in the range $20 \leq n \leq 100$ are adequate in most cases. Consequently, the computational cost and storage requirements are much smaller while obtaining a comparable accuracy than the global method.

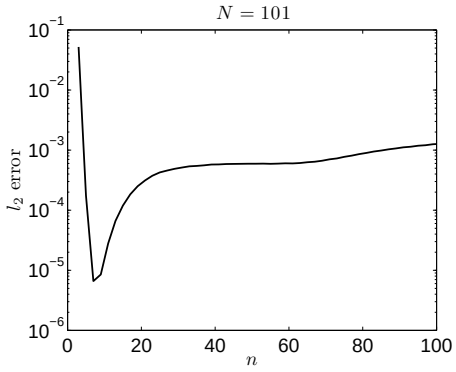


Figure 1.4: The l_2 error (left) and the shape parameter ε (right) versus the number of nodes N using a MQ RBF-FD approximation for the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$ keeping the condition number of A-matrix bounded in the range $10^{14} \leq \kappa \leq 10^{15}$.

Table 1.2: Absolute errors at different x-values for different stencil sizes n using a MQ RBF-FD approximation for the first derivative of $f = e^{\sin(\pi x)}$ in $[-1, 1]$ with $N = 101$ and $10^{14} \leq \kappa \leq 10^{15}$.

	x-value					
n	-1	-0.6	-0.2	0.2	0.6	1
5	$6.8 \cdot 10^{-5}$	$6.1 \cdot 10^{-7}$	$2.1 \cdot 10^{-6}$	$5.2 \cdot 10^{-6}$	$2.8 \cdot 10^{-5}$	$8.1 \cdot 10^{-5}$
8	$1.6 \cdot 10^{-7}$	$1.7 \cdot 10^{-7}$	$1.5 \cdot 10^{-7}$	$4.5 \cdot 10^{-7}$	$2.6 \cdot 10^{-7}$	$8.2 \cdot 10^{-6}$
11	$2.5 \cdot 10^{-5}$	$6.5 \cdot 10^{-8}$	$1.7 \cdot 10^{-7}$	$1.4 \cdot 10^{-7}$	$1.4 \cdot 10^{-7}$	$1.3 \cdot 10^{-5}$
15	$8.5 \cdot 10^{-5}$	$6.1 \cdot 10^{-8}$	$1.6 \cdot 10^{-7}$	$7.6 \cdot 10^{-8}$	$8.3 \cdot 10^{-8}$	$8.2 \cdot 10^{-5}$
20	$1.7 \cdot 10^{-4}$	$6.1 \cdot 10^{-8}$	$2.8 \cdot 10^{-8}$	$3.7 \cdot 10^{-7}$	$4.7 \cdot 10^{-7}$	$2.7 \cdot 10^{-4}$
25	$2.9 \cdot 10^{-4}$	$5.5 \cdot 10^{-8}$	$1.3 \cdot 10^{-7}$	$1.1 \cdot 10^{-7}$	$8.2 \cdot 10^{-8}$	$3.1 \cdot 10^{-4}$
30	$3.6 \cdot 10^{-4}$	$7.1 \cdot 10^{-8}$	$3.7 \cdot 10^{-8}$	$2.6 \cdot 10^{-7}$	$2.5 \cdot 10^{-7}$	$3.6 \cdot 10^{-4}$
40	$4.7 \cdot 10^{-4}$	$5.6 \cdot 10^{-8}$	$8.8 \cdot 10^{-9}$	$1.5 \cdot 10^{-7}$	$1.1 \cdot 10^{-7}$	$3.3 \cdot 10^{-4}$
50	$5.0 \cdot 10^{-4}$	$5.8 \cdot 10^{-8}$	$1.5 \cdot 10^{-8}$	$6.2 \cdot 10^{-8}$	$3.9 \cdot 10^{-8}$	$3.1 \cdot 10^{-4}$
101	$1.1 \cdot 10^{-3}$	$1.3 \cdot 10^{-7}$	$2.5 \cdot 10^{-9}$	$2.9 \cdot 10^{-9}$	$6.8 \cdot 10^{-8}$	$5.7 \cdot 10^{-4}$

Chapter 2

Multiquadric RBF-FD formulas

2.1 Outline

It is well known that the local RBF method lacks the spectral accuracy of the global method. However, the exact dependence of the error with the inter-nodal distance h, shape parameter c and stencil size n was unknown. We mention, though, that Ding *et.al.* [20] carried out numerical experiments using Poisson's equation on an equispaced grid to experimentally determine these dependencies. They found for the multiquadric an error estimate $O((h/c)^k)$, where k is a constant dependent on the stencil size ($k \approx 1.9$ for $6 \leq n \leq 9$, $n \approx 3.6$ for $9 < n \leq 27$, $n \approx 4.9$ for $27 < n \leq 34$).

The purpose of this chapter is to fill up this gap by analyzing the exact convergence properties of RBF-FD formulas. As in [20], we use multiquadrics as RBFs, where c is the shape parameter using Hardy's notation (1.3). The analysis will be extended for gaussians (GA) in Appendix A. Following the theory of Section 1.4, we consider RBF-FD formulas for first and second order derivatives in 1D (Sections 2.2 and 2.3) and for the Laplacian in 2D (Section 2.4), using equispaced nodes in all cases. We use Taylor series expansions in the limit $c \gg h$ to derive closed form expressions of the weighting coefficients and calculate for each stencil size n the corresponding local truncation errors of the approximations as function of h and c. These formulas allow us to establish the convergence properties of the method and derive analytical expressions for the optimal value of the shape parameter (Section 2.5). In Section 2.6 the results are extended to the case of non-equispaced nodes. The main conclusions of this chapter are finally summarized in Section 2.7.

2.2 First Derivative

Consider an RBF-FD approximation (1.17) to the first derivative using three equispaced nodes, i.e.

$$\hat{u}'(x_0) \,=\, \alpha_{-1}\, u(x_0 \,-\, h) \,+\, \alpha_0\, u(x_0) \,+\, \alpha_{+1}\, u(x_0 \,+\, h). \tag{2.1}$$

For simplicity, we have labeled the one dimensional weights following the notation

$$i = \begin{cases} -(n-1)/2, \ldots, (n-1)/2 & n \text{ odd} \\ -n/2, \ldots, n/2-1 & n \text{ even} \end{cases}.$$

Notice that in both cases the label for the stencil center is 0.

Imposing (2.1) to be exact for MQ, results in the linear system of equations

$$\begin{bmatrix} c & \sqrt{h^2+c^2} & \sqrt{4\,h^2+c^2} \\ \sqrt{h^2+c^2} & c & \sqrt{h^2+c^2} \\ \sqrt{4\,h^2+c^2} & \sqrt{h^2+c^2} & c \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \frac{h}{\sqrt{h^2+c^2}} \\ 0 \\ -\frac{h}{\sqrt{h^2+c^2}} \end{bmatrix} \tag{2.2}$$

whose solution is,

$$\begin{aligned} \alpha_0 &= 0, \\ \alpha_1 = -\alpha_{-1} &= -\frac{h}{\sqrt{c^2+h^2}\left(c-\sqrt{c^2+4h^2}\right)}. \end{aligned}$$

In the limit $c \gg h$,

$$\alpha_1 \,=\, -\,\alpha_{-1} \,=\, \frac{1}{2\,h} \,+\, \frac{h}{4\,c^2} \,+\, O\left(\frac{h^3}{c^4}\right), \qquad \alpha_0 \,=\, 0\,, \tag{2.3}$$

which coincides with the standard central difference approximation to the first derivative with a correction term of order h/c^2.

Including additional nodes simply leads to larger linear systems to determine the coefficients of the RBF-FD formulas for the first derivatives. Using a symbolic language (such as *Mathematica* or *Maple*) it is possible to derive the exact formulas for the coefficients. Nevertheless, the formula for six nodes computed with *Mathematica* is 45 pages long. In these cases it is useful to compute the Taylor series expression when $c \gg h$. Some results are shown in Table 2.1.

Table 2.1: RBF-FD weights for the first derivative

Three nodes	
α_0	0
$\alpha_{\pm1}$	$\mp\frac{h}{\sqrt{c^2+h^2}\left(c-\sqrt{c^2+4h^2}\right)}$
Four nodes	
α_0	$-\frac{\left(\sqrt{c^2+4h^2}+c\right)\left(3\sqrt{c^2+h^2}-\sqrt{c^2+9h^2}\right)}{8h\sqrt{c^2+h^2}\sqrt{c^2+4h^2}}$
α_{+1}	$\frac{c\left(\sqrt{c^2+4h^2}+c\right)+2h^2}{2h\sqrt{c^2+h^2}\sqrt{c^2+4h^2}}$
α_{-1}	$-\frac{c\sqrt{c^2+4h^2}+\sqrt{c^2+h^2}\sqrt{c^2+9h^2}+3h^2}{6h\sqrt{c^2+h^2}\sqrt{c^2+4h^2}}$
α_{+2}	$-\frac{-\frac{c\left(\sqrt{c^2+h^2}-3\sqrt{c^2+9h^2}\right)}{\sqrt{c^2+h^2}\sqrt{c^2+4h^2}}-\frac{\sqrt{c^2+9h^2}}{\sqrt{c^2+h^2}}+3}{24h}$
Five nodes	
α_0	0
$\alpha_{\pm1}$	$\pm\frac{2}{3h}\pm\frac{4h}{3c^2}\mp\frac{95h^3}{12c^4}+O\left(\frac{h^5}{c^6}\right)$
$\alpha_{\pm2}$	$\mp\frac{1}{12h}\mp\frac{2h}{3c^2}\pm\frac{10h^3}{3c^4}+O\left(\frac{h^5}{c^6}\right)$
Six nodes	
α_0	$-\frac{1}{3h}+\frac{240h}{77c^2}-\frac{23256h^3}{847c^4}+\frac{2482434h^5}{9317c^6}+O\left(\frac{h^7}{c^8}\right)$
α_{+1}	$+\frac{1}{h}-\frac{73h}{154c^2}+\frac{17253h^3}{6776c^4}-\frac{3152381h^5}{149072c^6}+O\left(\frac{h^7}{c^8}\right)$
α_{-1}	$-\frac{1}{2h}-\frac{887h}{308c^2}+\frac{42357h^3}{1936c^4}-\frac{57325819h^5}{298144c^6}+O\left(\frac{h^7}{c^8}\right)$
α_{+2}	$-\frac{1}{4h}-\frac{167h}{154c^2}+\frac{1323h^3}{121c^4}-\frac{974095h^5}{9317c^6}+O\left(\frac{h^7}{c^8}\right)$
α_{-2}	$+\frac{1}{20h}+\frac{647h}{770c^2}-\frac{17931h^3}{4235c^4}+\frac{1004893h^5}{46585c^6}+O\left(\frac{h^7}{c^8}\right)$
α_{+3}	$+\frac{1}{30h}+\frac{741h}{1540c^2}-\frac{248529h^3}{67760c^4}\frac{35877489h^5}{1490720c^6}+O\left(\frac{h^7}{c^8}\right)$

It is interesting to compute the local truncation errors resulting from these formulas. For instance, in the case above, introducing the values of the coefficients given by (2.3) into (2.1), and expanding $u(x_0+h)$ and $u(x_0-h)$ around x_0 results in

$$\begin{aligned}
\tau_3(x_0) &\equiv \hat{u}'(x_0) - u'(x_0) = \\
&= \frac{1}{2\,h}\left(1 + \frac{h^2}{2\,c^2}\right)\left[u(x_0 + h) - u(x_0 - h)\right] - u'(x_0) = \\
&= \frac{1}{2\,h}\left(1 + \frac{h^2}{2\,c^2}\right)\left[2\,h\,u'(x_0) + \frac{h^3}{3}\,u'''(x_0) + \ldots\right] - u'(x_0) = \\
&= \frac{h^2}{6}\left(u'''(x_0) + \frac{3}{c^2}u'(x_0)\right) + O\left(h^4 P_2(1/c^2)\right), \qquad (2.4)
\end{aligned}$$

which is second order in h, like the standard central difference formula, with a correction term of order $(h/c)^2$. The next term in the series expansion is fourth order in h and a polynomial of second order in $1/c^2$. We use the notation $O\left(h^m\,P_n(1/c^2)\right)$ to indicate that the terms that have been neglected are of order $h^m \sum_{i=0}^{n} \frac{a_i}{c^{2i}}$, where a_i are constants which depend on the derivatives and values of the particular function at x_0.

We can check these results numerically by computing the error in approximating the first derivative of $u = \exp\left(-x^2\right)$ at $x = 1$. Figure 2.1 shows the error as a function of c/h (left) and as a function of h (right). Both figures show that the numerical results (solid lines) closely agree with equation (2.4) (dot-dashed lines) until a critical value of c/h is reached ($c/h \approx 5000$) when the linear system (2.2) becomes ill conditioned and rounding errors deteriorate the accuracy of the solution. For small values of c/h, the contribution of the second term in (2.4), $\dfrac{h^2}{2\,c^2}u'(x_0)$, is dominant and the error shown in the left side of Figure 2.1 decreases as $(1/c)^2$. For large values of c/h, the contribution of the first term, $\dfrac{h^2}{6}\,u'''(x_0)$, is dominant and the error approaches a constant value. In the case of the error dependence with h, shown in the right side of the figure, the first term is dominant throughout.

Notice that the shape parameter c makes possible to minimize the approximation error given by equation (2.4). In fact, for this simple case the error is zero to leading order for $c^2 = -3\,u'(x_0)\,/\,u'''(x_0)$. If $u'(x_0)$ and $u'''(x_0)$ have opposite signs then c^2 is positive and there is a real positive value of c for which the error is zero (see Section 2.5).

We can repeat the same procedure for the case $n = 4$. After some algebra,

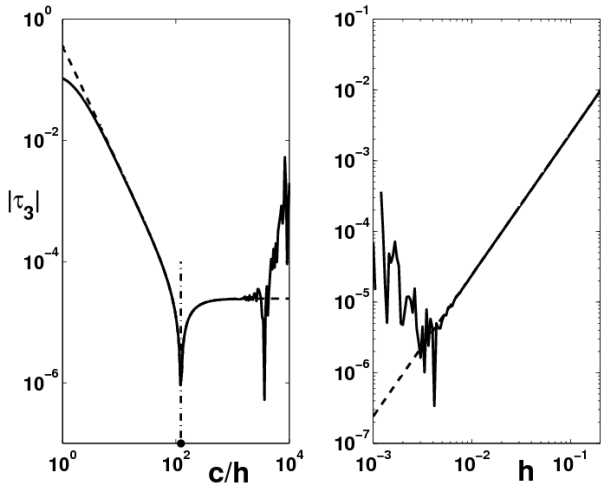


Figure 2.1: Error in approximation of first derivative with RBF-FD formula for three equispaced nodes. $u(x) = \exp(-x^2)$, $x = 1$. Left: c/h dependence for $h = 0.01$. Right: h dependence for $c = 10$. Dashed line equation (2.4).

the local truncation error is

$$\tau_4(x_0) = -\frac{h^3}{12}\left(u^{(IV)}(x_0) + \frac{12}{c^2}u''(x_0) - \frac{9}{c^4}u(x_0)\right) + O\left(h^4 P_2(1/c^2)\right). \quad (2.5)$$

Figure 2.2 shows the error in the approximation of the first derivative of $u = \exp(-x^2)$ at $x = 1$ using the RBF-FD formula for four equispaced nodes and compares it to the error given by equation (2.5). As was the case with three nodes, there is a critical value of the shape parameter c above which the system becomes ill conditioned leading to high errors. Notice also that for small values of c/h the contribution of the second term, $\dfrac{3\,h^3}{4\,c^4}$, is dominant and the error shown in the left side of Figure 2.2 decreases as $(1/c)^4$. For large values of c/h the contribution of the first term, $\dfrac{h^3}{12}\,u^{(IV)}(x_0)$, is dominant and the error approaches a constant value. There is an intermediate region around $c/h \approx 10^2$, where the three terms are comparable.

The corresponding results for $n = 5$ and $n = 6$ are

$$\tau_5(x_0) = -\frac{h^4}{30}\left(u^{(V)}(x_0) + \frac{40}{c^2}u'''(x_0) + \frac{75}{c^4}u'(x_0)\right) + O\left(h^6 P_3(1/c^2)\right)$$

and

$$\tau_6(x_0) = \frac{h^5}{60}\left(u^{(VI)}(x_0) + \frac{555}{7c^2}u^{(IV)}(x_0) + \frac{3825}{7c^4}u''(x_0) - \frac{2475}{7c^6}u(x_0)\right) + O\left(h^6 P_3(1/c^2)\right).$$

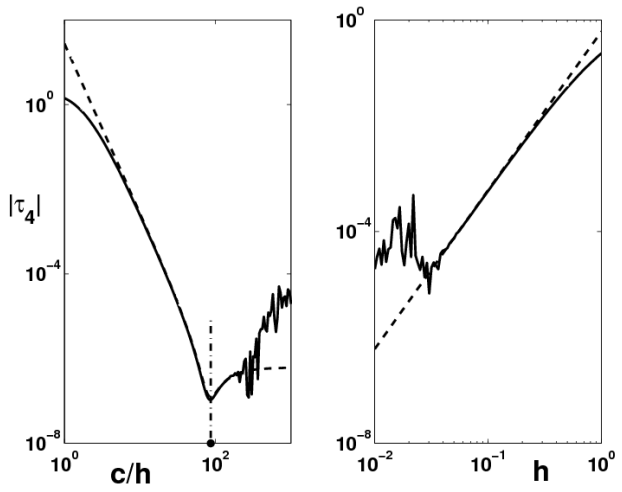


Figure 2.2: Error in approximation of first derivative with RBF-FD formula with four equispaced nodes. $u(x) = \exp(-x^2)$, $x = 1$. Left: c/h dependence for $h = 0.01$. Right: h dependence for $c = 10$. Dashed line equation (2.5).

Thus, the errors of the RBF-FD formulas which approximate the first derivative with n nodes can be written to leading order as

$$\tau_n(x_0) \;=\; h^{n-1}\, P_{(n+k-1)/2}(1/c^2) \tag{2.6}$$

where $k = 0$ if n odd and $k = 1$ if n even. The polynomial is

$$P_{(n+k-1)/2}(1/c^2) = \sum_{m=0}^{(n+k-1)/2} \frac{A_m}{c^{2\,m}}\, u^{(n-2m)}(x_0),$$

where A_m are constants which depend on n. For the smaller values of c/h the error behaves as

$$\tau_n(x_0) = \begin{cases} O(h/c)^{n-1}\, u'(x_0) & \text{if } n \text{ odd} \\ O(h^{n-1}/c^n)\, u(x_0) & \text{if } n \text{ even} \end{cases}, \tag{2.7}$$

while for large values of c/h it approaches a value independent of c. This value coincides with the corresponding standard finite difference error. For intermediate values of c/h some of the other terms might become dominant, depending on the particular function u and the value of h used (see for instance Figure 2.3).

2.3 Second Derivative

Analogously, we derive the RBF-FD approximation to the second derivative using three equispaced nodes. In this case

$$\frac{d^2u}{dx^2}(x_0) \;=\; \alpha_{-1}\,u(x_0 \,-\, h) \,+\, \alpha_0\,u(x_0) \,+\, \alpha_1\,u(x_0 \,+\, h)\,. \tag{2.8}$$

Similar to Section 2.2, equation (2.8) is imposed to be exact for MQ centered at $\{x_0 - h,\, x_0,\, x_0 + h\}$, which after solving the resulting linear system leads to the weights

$$\alpha_0 \;=\; -\frac{2 \,+\, \left(\dfrac{h^2}{c^2} \,+\, 2\right)\sqrt{1 \,+\, 4\,\dfrac{h^2}{c^2}} \,+\, 3\,\dfrac{h^2}{c^2}}{2\,h^2\left(1 \,+\, \dfrac{h^2}{c^2}\right)},$$

$$\alpha_1 \,=\, \alpha_{-1} \;=\; \frac{2 \,+\, \left(\dfrac{h^2}{c^2} \,+\, 2\right)\sqrt{1 \,+\, 4\,\dfrac{h^2}{c^2}} \,+\, 5\,\dfrac{h^2}{c^2} \,+\, 2\,\dfrac{h^4}{c^4}}{4\,h^2\left(1 \,+\, \dfrac{h^2}{c^2}\right)^{3/2}}.$$

In the limit $c \gg h$,

$$\alpha_0 \;=\; -\frac{2}{h^2} \,-\, \frac{2}{c^2} \,+\, O\left(\frac{h^2}{c^4}\right), \tag{2.9}$$

$$\alpha_1 \,=\, \alpha_{-1} \;=\; +\frac{1}{h^2} \,+\, \frac{1}{c^2} \,+\, O\left(\frac{h^2}{c^4}\right), \tag{2.10}$$

which again coincides with the standard central difference approximation to the second derivative with a correction term of order $1/c^2$. Table 2.2 shows the corresponding RBF-FD weights for other values of n.

Introducing the values of the coefficients given by (2.10) into (2.8), and expanding $u(x_0+h)$ and $u(x_0-h)$ around x_0 we find the corresponding local truncation error

$$\tau_3(x_0) \,=\, \frac{h^2}{12}\left(u^{(IV)}(x_0) + \frac{12}{c^2}u''(x_0) - \frac{9}{c^4}u(x_0)\right), \tag{2.11}$$

where the terms neglected are of order $O\left(h^4 P_3(1/c^2)\right)$. The same error dependence to leading order is obtained for $n = 4$. The local truncation error for $n = 5$ is

$$\tau_5(x_0) \,=\, -\frac{h^4}{90}\left(u^{(VI)}(x_0) + \frac{555}{7c^2}u^{(IV)}(x_0) + \frac{3825}{7c^4}u''(x_0) - \frac{2475}{7c^6}u(x_0)\right), \tag{2.12}$$

Table 2.2: RBF-FD second derivative

Three nodes	
α_0	$-\frac{2c^3+2c^2\sqrt{c^2+4h^2}+h^2\sqrt{c^2+4h^2}+3ch^2}{2c^3h^2+2ch^4}$
$\alpha_{\pm1}$	$\frac{h^2\left(2c^2+h^2\right)\left(c-\sqrt{c^2+4h^2}\right)}{2c(c^2+h^2)^{3/2}\left(c^3-c^2\sqrt{c^2+4h^2}-h^2\sqrt{c^2+4h^2}+3ch^2\right)}$
Four nodes	
α_0	$-\frac{2}{h^2}-\frac{11}{2c^2}+\frac{47h^2}{2c^4}+O\left(\frac{h^4}{c^6}\right)$
α_{+1}	$\frac{1}{h^2}+\frac{9}{2c^2}-\frac{137h^2}{8c^4}+O\left(\frac{h^4}{c^6}\right)$
α_{-1}	$\frac{1}{h^2}+\frac{13}{6c^2}-\frac{223h^2}{24c^4}+O\left(\frac{h^4}{c^6}\right)$
α_{+2}	$-\frac{7}{6c^2}+\frac{13h^2}{6c^4}+O\left(\frac{h^4}{c^6}\right)$
Five nodes	
α_0	$-\frac{5}{2h^2}-\frac{37}{7c^2}+\frac{1345h^2}{49c^4}-\frac{117433h^4}{686c^6}+O\left(\frac{h^6}{c^8}\right)$
$\alpha_{\pm1}$	$\frac{4}{3h^2}+\frac{74}{21c^2}-\frac{1595h^2}{98c^4}+\frac{399557h^4}{4116c^6}+O\left(\frac{h^6}{c^8}\right)$
$\alpha_{\pm2}$	$-\frac{1}{12h^2}-\frac{37}{42c^2}+\frac{125h^2}{49c^4}-\frac{39173h^4}{4116c^6}+O\left(\frac{h^6}{c^8}\right)$
Six nodes	
α_0	$-\frac{5}{2h^2}-\frac{713}{77c^2}+\frac{77849h^2}{847c^4}-\frac{19041359h^4}{18634c^6}+O\left(\frac{h^6}{c^8}\right)$
α_{+1}	$\frac{4}{3h^2}+\frac{1732}{231c^2}-\frac{110323h^2}{1694c^4}+\frac{17829520h^4}{27951c^6}+O\left(\frac{h^6}{c^8}\right)$
α_{-1}	$\frac{4}{3h^2}+\frac{1273}{231c^2}-\frac{40965h^2}{847c^4}+\frac{117219947h^4}{223608c^6}+O\left(\frac{h^6}{c^8}\right)$
α_{+2}	$-\frac{1}{12h^2}-\frac{1325}{462c^2}+\frac{9505h^2}{847c^4}-\frac{2091055h^4}{111804c^6}+O\left(\frac{h^6}{c^8}\right)$
α_{-2}	$-\frac{1}{12h^2}-\frac{2953}{2310c^2}+\frac{31131h^2}{4235c^4}-\frac{34255187h^4}{559020c^6}+O\left(\frac{h^6}{c^8}\right)$
α_{+3}	$\frac{153}{385c^2}+\frac{25463h^2}{8470c^4}-\frac{20995257h^4}{372680c^6}+O\left(\frac{h^6}{c^8}\right)$

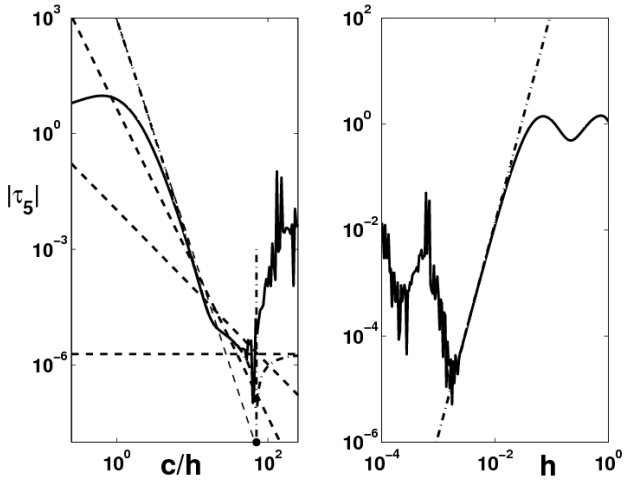


Figure 2.3: Error in approximation of second derivative with RBF-FD formula with five equispaced nodes. Left: c/h dependence for $h = 0.04$. Right: h dependence for $c = 0.1$. Dot-dashed line equation (2.12). Dashed lines: each of the terms in equation (2.12).

where the terms neglected are of order $O\left(h^6 P_4(1/c^2)\right)$. The same dependence to leading order is obtained for $n = 6$.

As an example, Figure 2.3 shows the error in the approximation of the second derivative of $u = \exp(-x^2)$ at $x = 1$ using the RBF-FD formula for five equispaced nodes and compares it to the error given by equation (2.12). Again, the error predicted by the equation (dot-dashed lines) closely agrees with the actual numerical error (solid lines) until a critical value of the shape parameter is reached above which the system becomes ill-conditioned. Also shown in the left side of the figure (thin dashed lines), are the contributions of each one of the four terms appearing in equation (2.12) to the total error $\tau_5(x_0)$. For the smaller values of c/h the contribution of the last term is dominant and, therefore, the error decreases as $(1/c)^6$. For larger values of c/h the contribution of the first term is dominant and, therefore, $\tau_5(x_0)$ approaches a constant (this is not observed in the numerical results because those large values of c/h lie in the ill- conditioned region). For intermediate values of c/h, there is a region where the second term is dominant and the error decreases as $(1/c)^2$.

In general, the errors of the RBF-FD formulas which approximate the second derivative with n nodes can be written to leading order as

$$\tau_n(x_0) \;=\; h^{n+k-2}\, P_{(n+k)/2}(1/c^2), \tag{2.13}$$

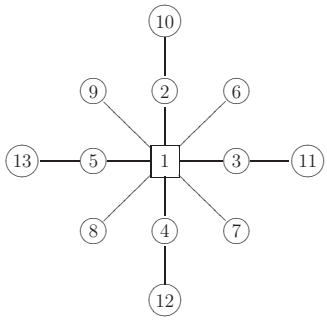


Figure 2.4: Order of nodes in 2-D equispaced stencil.

where $k = 0$ if n even and $k = 1$ if n odd. The polynomial is

$$P_{(n+k)/2}(1/c^2) = \sum_{m=0}^{(n+k)/2} \frac{A_m}{c^{2\,m}}\, u^{(n+k-2m)}(x_0),$$

where A_m are constants which depend on n.

2.4 Two-dimensional RBF-FD Formulas

In this section we use the same procedure of the previous Section to derive RBF-FD formulas for the Laplacian. We compute the limit of these formulas for $c \gg h$, and perform a Taylor expansion of the error in powers of h.

2.4.1 Laplacian

To compute the errors for the RBF-FD formulas of the Laplacian we can proceed as in the previous section by computing the exact values of the coefficients with a symbolic program (*Mathematica*) and by using these values to perform a Taylor series expansion for the corresponding errors. We take the nodes from a regular, equispaced grid, following the same order convention used in reference [109] which is shown in Figure 2.4.

For instance, in the case of the RBF-FD formula for five nodes, the coefficients in the limit $c \gg h$ are

$$\begin{aligned}
\alpha_1 &= -\frac{4}{h^2} - \frac{10}{3\,c^2} + \frac{43h^2}{9c^4} + O\left(\frac{h^4}{c^6}\right), \\
\alpha_i &= \frac{1}{h^2} + \frac{5}{6\,c^2} - \frac{107h^2}{72c^4} + O\left(\frac{h^4}{c^6}\right), \qquad i = 2, \ldots, 5\,,
\end{aligned}$$

and the error of the approximation is given by[1]

$$\begin{aligned}\tau_5(\mathbf{x}_0) &= \tfrac{h^2}{12}\left[\left(u^{(4,0)}(\mathbf{x}_0)+u^{(0,4)}(\mathbf{x}_0)\right)+\right.\\ &\qquad\left.+\tfrac{10}{c^2}\left(u^{(2,0)}(\mathbf{x}_0)+u^{(0,2)}(\mathbf{x}_0)\right)-\tfrac{14}{c^4}\,u(\mathbf{x}_0)\,\right],\end{aligned} \tag{2.14}$$

where the terms neglected are of order $O\left(h^4\,P_3(1/c^2)\right)$. For the six nodes approximation, the RBF-FD coefficients are

$$\begin{aligned}\alpha_1 &= -\tfrac{4}{h^2}-\tfrac{10}{3c^2}-\tfrac{5h^2}{9c^4}+O\left(\tfrac{h^4}{c^6}\right),\\ \alpha_{2,3} &= \tfrac{1}{h^2}+\tfrac{5}{6c^2}+\tfrac{277h^2}{72c^4}+O\left(\tfrac{h^4}{c^6}\right),\\ \alpha_{4,5} &= \tfrac{1}{h^2}+\tfrac{5}{6c^2}-\tfrac{107h^2}{72c^4}+O\left(\tfrac{h^4}{c^6}\right),\\ \alpha_6 &= \tfrac{16h^2}{3c^4}+O\left(\tfrac{h^4}{c^6}\right).\end{aligned}$$

Notice that these coefficients differ by an amount of order $O\left(\frac{h^2}{c^4}\right)$ from those obtained in the five nodes approximation. This results in a small change of the error, so that

$$\tau_6(\mathbf{x}_0) = \tau_5(\mathbf{x}_0) + O\left(\frac{h^4}{c^4}\right).$$

Analogously, the RBF-FD coefficients for $n=7$ are up to order $O\left(\dfrac{h^4}{c^6}\right)$

$$\begin{aligned}\alpha_1 &= -\tfrac{4}{h^2}-\tfrac{6}{c^2}+\tfrac{40h^2}{3c^4}, & \alpha_{2,4} &= \tfrac{1}{h^2}+\tfrac{13}{6c^2}-\tfrac{335h^2}{72c^4},\\ \alpha_3 &= \tfrac{1}{h^2}+\tfrac{7}{2c^2}-\tfrac{59h^2}{8c^4}, & \alpha_5 &= \tfrac{1}{h^2}+\tfrac{5}{6c^2}-\tfrac{139h^2}{72c^4},\\ \alpha_{6,7} &= -\tfrac{4}{3c^2}+\tfrac{37h^2}{18c^4}.\end{aligned}$$

In this case, it also results in a small change of the error such that

$$\tau_7(\mathbf{x}_0) = \tau_5(\mathbf{x}_0) + O\left(\frac{h^3}{c^2}\right).$$

[1] $u^{(m,n)}$ denotes the partial derivative of function u with respect to x, m times and respect to y, n times

For $n = 9$ the symmetry is restored, being the coefficients symmetrical

$$\begin{aligned}
\alpha_1 &= -\tfrac{372}{77h^2} - \tfrac{35198}{5929c^2} + \tfrac{7032819h^2}{456533c^4} + O\left(\tfrac{h^4}{c^6}\right), \\
\alpha_{2,3,4,5} &= \tfrac{109}{77h^2} + \tfrac{29423}{11858c^2} - \tfrac{22158463h^2}{3652264c^4} + O\left(\tfrac{h^4}{c^6}\right), \\
\alpha_{6,7,8,9} &= -\tfrac{16}{77h^2} - \tfrac{5912}{5929c^2} + \tfrac{933785h^2}{456533c^4} + O\left(\tfrac{h^4}{c^6}\right),
\end{aligned}$$

and the error

$$\begin{aligned}
\tau_9(\mathbf{x}_0) = \tfrac{h^2}{12}\Big[\big(u^{(4,0)}(\mathbf{x}_0) - \tfrac{192}{77}\,u^{(2,2)}(\mathbf{x}_0) + u^{(0,4)}(\mathbf{x}_0)\big) + \\
+ \tfrac{900}{154\,c^2}\,\big(u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0)\big) - \tfrac{90}{11\,c^4}\,u(\mathbf{x}_0)\Big],
\end{aligned} \tag{2.15}$$

where the terms neglected are of order $O\left(h^4\,P_3(1/c^2)\right)$.

For $n > 9$ the computational requirements to obtain closed form solutions for the coefficients and for the error using *Mathematica* are too high. However, it is possible to derive numerically the dependence of the error with h, c and with the partial derivatives of the function by choosing appropriately the function to approximate. For instance, to determine the coefficient of $u^{(1,2)}$ in the Laplacian with n nodes, one can use the corresponding RBF-FD formula to compute numerically the Laplacian of $u(\mathbf{x}) = x\,y^2$ at $\mathbf{x}_1 = (0,\,0)$ for different values of h and c. Fitting the results to a power dependence with h and c determines the exact form of the coefficient of $u^{(1,2)}$. In this way, we derive formulas for the error for any number of nodes n. The most interesting result is for $n = 13$ since the symmetries with respect to x and y increase the accuracy of the approximation. In this case, the error is

$$\begin{aligned}
\tau_{13}(\mathbf{x}_0) \approx -\tfrac{h^4}{90}\,\big[u^{(6,0)}(\mathbf{x}_0) + u^{(0,6)}(\mathbf{x}_0)\big] + \\
-\frac{h^4}{c^2}\,\big[0.93\,u^{(4,0)}(\mathbf{x}_0) - 0.5\,u^{(2,2)}(\mathbf{x}_0) + 0.93\,u^{(0,4)}(\mathbf{x}_0)\big] + \\
-\frac{h^4}{c^6}\,4.4\,\big[u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0)\big] + 5.2\,u(\mathbf{x}_0).
\end{aligned} \tag{2.16}$$

For $c \gg h$, the general behavior of the error dependence of the n nodes RBF-FD formula for the Laplacian is to leading order

$$\tau_n(\mathbf{x}_0) = h^s\,P_{s/2+1}(1/c^2), \tag{2.17}$$

where s is the smallest even number that satisfies

$$(s-1)^2 + 4 \le n \le (s+1)^2 + 3$$

and the polynomial is

$$P_{s/2+1}(1/c^2) \;=\; \sum_{m=0}^{(s/2+1)} \; \sum_{r=0}^{(s/2+1-m)} \; \frac{A_{m,r}}{c^{2\,m}}\, u^{(s+2-2\,(m+r),2r)}(\mathbf{x}_0),$$

where $A_{m,r}$ are constants which depend on n.

2.4.2 Numerical experiments

In this Section we carry out numerical experiments to compute the error of the RBF-FD formulas for the Laplacian, and use them to check the analytical results derived in the previous section. As a first experiment we use the same functions analyzed in reference [20]. Figure 2.5 shows the dependence of the relative error versus the number nodes for the case $c = 0.2$ and $h_1 = 0.025$ (left side) and $h_2 = 0.01$ (right side) (h_1 and c are the parameters chosen in Figure 5 of reference [20]). The functions used are

$$\begin{aligned}
u_1 \;&=\; \frac{3}{4}\, e^{-\frac{(9\,x-2)^2+(9\,y-2)^2}{4}} \;+\; \frac{3}{4}\, e^{-\frac{(9\,x+1)^2}{49} - \frac{(9\,y+1)}{10}} \;+\; \frac{1}{2}\, e^{-\frac{(9\,x-7)^2+(9\,y-3)^2}{4}} \\
&\quad - \frac{2}{10}\, e^{-(9\,x-4)^2 - (9\,x-7)^2}\,, \\
u_2 \;&=\; \left(1 - \frac{x}{2}\right)^6 \left(1 - \frac{y}{2}\right)^6 \;+\; 1000\,(1-x)^3\, x^3\, (1-y)^3\, y^3 \;+ \\
&\quad + y^6 \left(1 - \frac{x}{2}\right)^6 \;+\; x^6 \left(1 - \frac{y}{2}\right)^6\,, \\
u_3 \;&=\; \sin(\pi\, x)\, \sin(\pi\, y)\,, \\
u_4 \;&=\; x^2 \;+\; y^2\,.
\end{aligned}$$

For clarity of the figure we do not include the results of the analytical expressions of the error from the sections above, but it should be remarked that they closely agree with the numerical results. Notice the existence of plateaus where the errors are approximately constant separated by transition regions where the errors decrease rapidly. This is the same behavior shown in Figure 5 of reference [20], although both results are not identical. In fact, Figure 2.5 shows the relative local truncation error in approximating the Laplacian with RBF-FD formulas at a specific location $\mathbf{x}_0$, while Figure 5 of reference [20] shows the infinity norm of the relative global error in the solution of Poisson equation with the local RBF method. However, both should behave similarly as it is stated in Section 3.2.

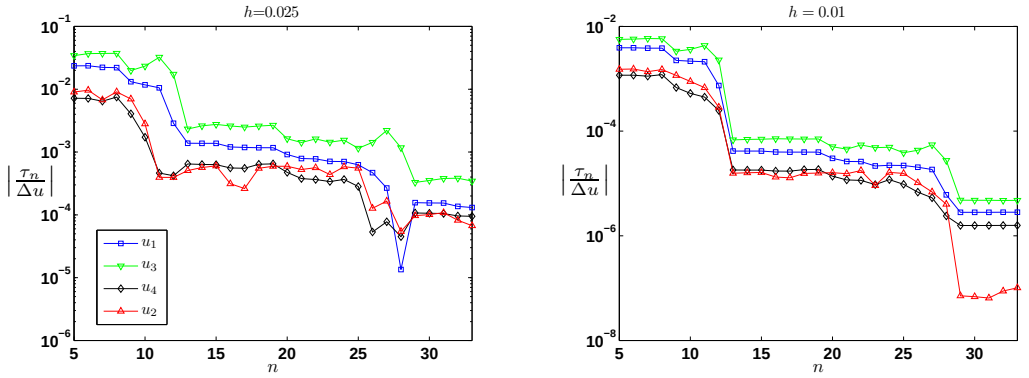


Figure 2.5: Relative error in the approximation of the Laplacian with RBF-FD formula as a function of the number of nodes n. $\mathbf{x}_0 = (0.1, 0.2)$, $c = 0.2$. Left; $h = 0.025$. Right; $h = 0.01$.

The behavior observed in Figure 2.5 can be better understood by considering the error dependence with h shown in Figure 2.6 for function u_1. The lines in blue intersect the errors of function u_1 at $h_1 = 0.025$ and $h_2 = 0.01$. The intersections are marked with squares and correspond with those of Figure 2.5. This figure is similar to Figure 7 of reference [20], and shows very similar behavior. Notice that if $h \ll c$, the error is $O(h^2)$ for $n = 5 - 12$, $O(h^4)$ for $n = 13 - 28$ and $O(h^6)$ for $n = 29 - 33$. This is the expected error dependence according to equation (2.17). If $h = O(c)$ terms of higher order in h which are neglected in equation (2.17) become important and introduce a correction in the results. This is the reason why plateaus in Figure 2.5 are much more constant for h_2 (right) than for h_1 (left). y axes and the correction is even greater.

Finally, to analyze the dependence of the error with shape parameter c, we consider the function

$$u(x,y) = \exp\left[-\left(x - \frac{1}{4}\right)^2 - \left(y - \frac{1}{2}\right)^2\right] \cos(2\,\pi\,y)\,\sin(\pi\,x)\,, \quad (2.18)$$

which was used by Wright and Fornberg [109] in their analysis of the solution of elliptic PDEs with RBF-FD and RBF-HFD formulas. Figure 2.7 shows the error as a function of c for different values of h. It shows a very similar behavior to Figure 2 of reference [109], where the infinity norm of the global error in the solution of Poisson equation is computed. As before, it should be remarked that both results should not be identical since Figure 2.7 shows the local truncation error in approximating the Laplacian at a specific location

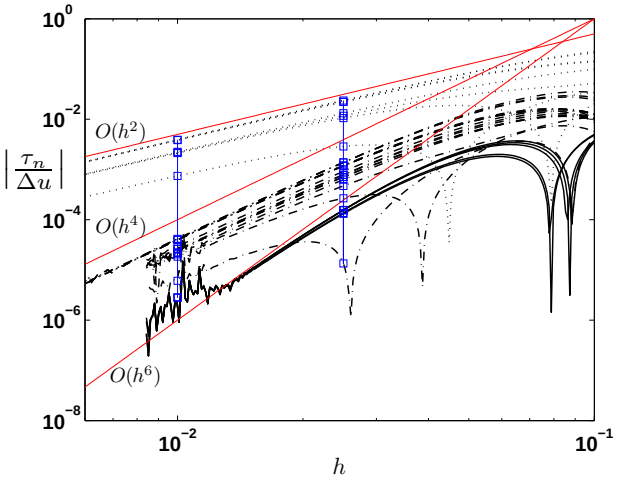


Figure 2.6: Relative error in the approximation of the Laplacian of function u_1 with RBF-FD formula as a function of h. $\mathbf{x}_0 = (0.1, 0.2)$, $c = 0.2$. $n = 5-12$ dotted lines, $n = 13-28$ dot-dashed lines, $n = 29-33$ solid lines.

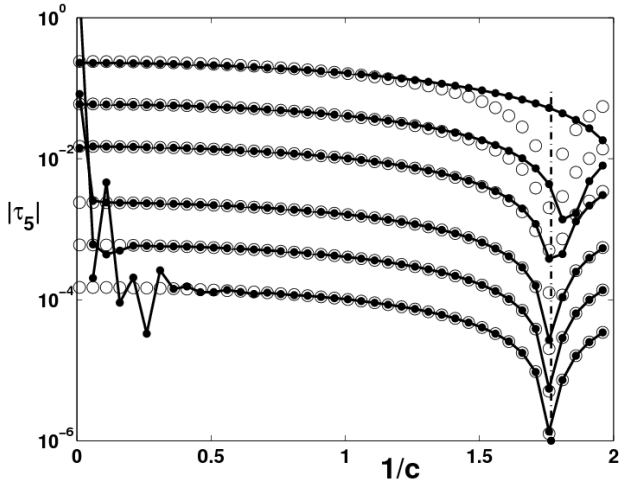


Figure 2.7: Error in approximation of Laplacian at $\mathbf{x}_0 = (0, 0)$ with $n = 5$ RBF-FD formula as a function of the inverse of the shape parameter $1/c$. From top to bottom, $h = 0.2$, $h = 0.1$, $h = 0.05$, $h = 0.02$, $h = 0.01$, $h = 0.005$. $\bullet$, numerical results. $\circ$, equation (2.14).

$\mathbf{x}_0$, while Figure 2 of reference [109] shows the infinity norm of the global error in the solution of Poisson equation. Notice that there is a value of the shape parameter for which the error is minimum. This value is approximately constant except for large values of h. Notice also that for c large and h small the resulting linear system becomes ill-conditioned and rounding error deteriorates the accuracy of the solution.

2.5 Optimal Shape Parameter

Several observations regarding the dependence of the error of the RBF-FD formulas with respect to shape parameter c are readily apparent from Figures 2.1 to 2.3 and 2.7:

- The error decreases with increasing c as some power which depends on the value of c/h.
- For large values of c, the conventional finite difference formulas are recovered as it was shown in [22, 44], and the error approaches a constant value which is the error of conventional finite differences.
- There is a range of values of c for which the error of the RBF-FD formulas is smaller than the error of conventional finite differences.
- There is an *optimal* value of the shape parameter for which the error is minimum.

Notice also that the optimal c^* is either a value for which $d\tau_n/dc$ is zero (Figure 2.2) or a value at which $\tau_n = 0$ (Figures 2.1, 2.3 and 2.7).

Since we have derived closed form expressions for the error of RBF-FD formulas, it is possible to compute in each case the optimal shape parameter c^* provided that the value of the function and its derivatives are known. Equations (2.6), (2.13) and (2.17) have the general form

$$\tau_n(x_0) \;\approx\; h^s \sum_{m=0}^{M} \frac{a_m(x_0)}{c^{2\,m}}\,, \tag{2.19}$$

where a_m are constants which depend on the derivatives and values of the particular function at x_0. Denoting $z \,=\, 1\,/\,c^2$, the optimal shape parameter is obtained from the positive real roots of the polynomials

$$a_1 \,+\, 2\,a_2\,z \,+\, \ldots \,+\, M\,a_M\,z^{M-1} \,=\, 0\,, \tag{2.20}$$

which implies $d\tau_n/dc = 0$, or

$$a_0 + a_1\, z + a_2\, z^2 + \ldots + a_M\, z^M = 0\,, \tag{2.21}$$

which implies $\tau_n = 0$. Solution of these two polynomials results in $2M - 1$ roots for $z = 1/c^2$. It is important to remark that the optimal shape parameter c^* only depends on the value of the function and its derivatives at the node. Therefore, to first order, it is independent of the mesh size h. For larger values of h there is a correction term of order $O(h)$.

For instance, let us consider the RBF-FD approximation of the second derivative of $u = \exp(-x^2)$ at $x = 1$ with five equispaced nodes, which is shown in Figure 2.3. The coefficients of the polynomials are given by (2.12), so that

$$\begin{aligned} a_0 &= -\tfrac{1}{90}u^{(VI)}(x_0), & a_1 &= -\tfrac{37}{42}u^{(IV)}(x_0), \\ a_2 &= -\tfrac{85}{14}u''(x_0), & a_3 &= \tfrac{55}{14}u(x_0). \end{aligned}$$

In this case, the two roots of the first polynomial (2.20) are complex ($\approx 1.03 \pm 0.658\, i$), and the three roots of the second polynomial (2.21) are two complex ($\approx 1.482 \pm 1.383\, i$) and one real, $z \approx 0.1266$. Thus, the optimal shape parameter is $c^* \approx 1/\sqrt{0.1266} \approx 2.81$, which is shown as a vertical dash-dotted line in Figure 2.3. The optimal shape parameters for the first derivative using three and four nodes can be analogously computed ($c^* \approx 1.2247$ and $c^* \approx 0.8666$, respectively) and are also shown with vertical lines in Figures 2.1 and 2.2.

In the case of the $n = 5 - 7$ Laplacian RBF-FD formula,

$$\begin{aligned} a_0 &= \tfrac{1}{12}\left(u^{(4,0)}(\mathbf{x}_0) + u^{(0,4)}(\mathbf{x}_0)\right), \\ a_1 &= \tfrac{5}{6}\left(u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0)\right), \\ a_2 &= -\tfrac{7}{6}u(\mathbf{x}_0). \end{aligned}$$

The solution of equations (2.20) and (2.21) in terms of the derivatives are,

$$(c^*)^2 = \frac{14\, u(\mathbf{x}_0)}{5\, d_2} \tag{2.22}$$

$$(c^*)^2 = \frac{14\, u(\mathbf{x}_0)}{5\, d_2 \pm \sqrt{25\, d_2^2 + 14\, d_4\, u(\mathbf{x}_0)}} \tag{2.23}$$

where

$$d_2 = u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0) \quad \text{and} \quad d_4 = u^{(4,0)}(\mathbf{x}_0) + u^{(0,4)}(\mathbf{x}_0).$$

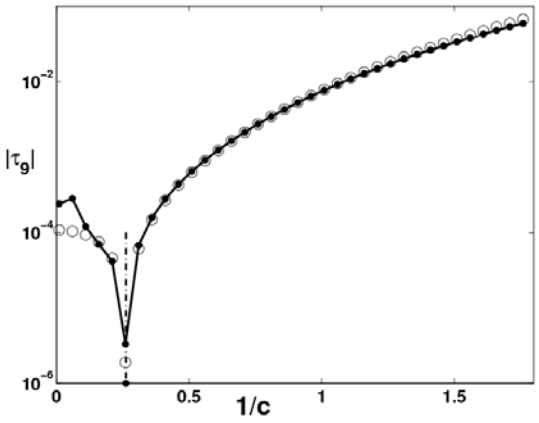


Figure 2.8: Error in approximation of Laplacian at $\mathbf{x}_0 = (0, 0)$ with $n = 9$ RBF-FD formula as a function of the inverse of the shape parameter $1/c$. $h = 0.1$. $\bullet$; numerical results. $\circ$; equation(2.15).

Thus, for the five nodes RBF-FD approximation to the Laplacian of function (2.18) at $\mathbf{x}_0 = (0, 0)$,

$$a_0 \approx -5.9835, \quad a_1 \approx 2.2984\,, \quad \text{and} \quad a_2 = 0.$$

Solution of equation (2.21) gives $z = -a_0/a_1 \approx 3.1239$, and the optimal shape parameter is $1/c^* \approx 1.7675$, which is shown in Figure 2.7.

As a last example we consider the function

$$u(\mathbf{x}) = \frac{25}{25 + (x - 0.2)^2 + 2\,y^2}\,, \tag{2.24}$$

which is the solution of the problem described in Section 5.2 of reference [109]. Figure 2.8 shows the error in approximation of the Laplacian with the nine nodes RBF-FD formula as a function of the inverse of the shape parameter. This dependence is very similar to that observed in Figure 4 of reference [109]. Also shown is the optimal value of the shape parameter which, using equation (2.15), results in $1/c^* \approx 0.2617$.

The problem of how to select appropriate values for the shape parameter has been of primary concern both from the theoretical and from the applications point of view. For the global RBF method [55, 56], it has been often assumed that the value of the shape parameter c should vary linearly with node spacing h. For instance, for interpolation problems, Hardy [50] suggests the use of $c = 0.815\,d$, where d is the average distance to the nearest neighbor ($d = h$ for equispaced nodes). Franke [46] on the other hand recommends

$c = 1.25\,D/\sqrt{N}$, where D is the diameter of the smallest circle containing all data points ($c = 1.25\sqrt{2}\,h$ for equispaced nodes). Other authors proposed techniques to select good values of the shape parameter [9, 65, 83]. With regards to the solution of PDEs, the work of Huang *et.al.* [53] using arbitrary precision computations, is of particular relevance. From their numerical results they derive a formula for the error dependence on shape parameter c and nodal spacing h. From this formula they obtain the optimal value of the shape parameter that minimizes the error; $c = -\log\lambda/(3\,a\,h)$, where a and λ are constants that depend of the problem.

However, our results show that the value of c is independent of h to leading order. Nevertheless, it should be pointed out that, in practical applications, the node density is often increased (h decreased) in regions where the solution varies rapidly. In these boundary layer type regions, the solution varies in small characteristic lengths ($l \ll L$). Thus, $d_2 = O(L/l)^2$, $d_4 = O(L/l)^4$ and, therefore, from (2.22)-(2.23) the optimal shape parameter is $c^* = O(l/L)$. In those regions, therefore, the shape parameter should be taken small not because h is small, but because the solution varies rapidly.

2.6 Unstructured nodes

In this Section we extend our results to the relevant case of unstructured nodes. For instance, in the case of three non-equispaced nodes $\{x_0-h,\, x_0,\, x_0+\lambda h\}$, the coefficients of the RBF-FD formula for the first derivative in the limit $c \gg h$ are

$$\begin{aligned}
\alpha_{-1} &= -\frac{\lambda}{h(\lambda+1)} - \frac{h\lambda^2}{2c^2(\lambda+1)} + \frac{h^3\lambda(2\lambda(2\lambda(\lambda+1)+1)-1)}{8c^4(\lambda+1)} + O\left(\frac{h^5}{c^6}\right),\\
\alpha_0 &= \frac{\lambda-1}{h\lambda} + \frac{h(\lambda-1)}{2c^2} - \frac{h^3\,(\lambda^3-1)}{2c^4} + O\left(\frac{h^5}{c^6}\right),\\
\alpha_1 &= \frac{1}{h\lambda(\lambda+1)} + \frac{h}{c^2(2\lambda+2)} + \frac{h^3(\lambda((\lambda-2)\lambda-4)-4)}{8c^4(\lambda+1)} + O\left(\frac{h^5}{c^6}\right),
\end{aligned}$$

which coincides with the standard 3-node finite difference approximation to the first derivative with a correction term of order $O\,(h^2/c^2)$. Also notice that for $\lambda = 1$ we recover the results of equation (2.3). The corresponding local truncation error is

$$\begin{aligned}
\tau_3(x_0) &= \lambda\,\tfrac{h^2}{6}\,u'''(x_0) + \lambda\,\tfrac{h^2}{2\,c^2}\,u'(x_0) + \lambda\,(\lambda-1)\,\tfrac{h^3}{24}\,u^{(IV)}(x_0) + \\
&\quad + \lambda\,(\lambda-1)\,\tfrac{h^3}{8\,c^4}\,u(x_0) + O\,(h^4\,P_2(1/c^2))\,,
\end{aligned} \tag{2.25}$$

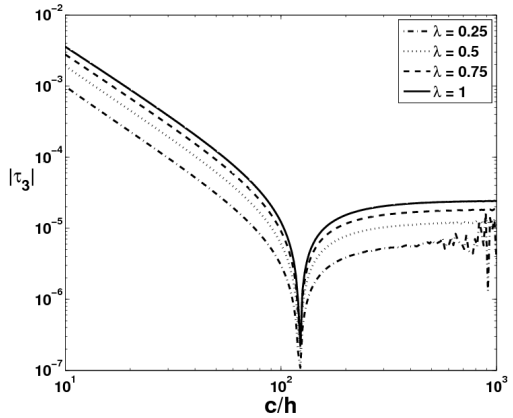


Figure 2.9: Error in approximation of first derivative with RBF-FD formula for three non-equispaced nodes as a function of c/h. $u(x) = \exp(-x^2)$, $x = 1$, $h = 0.01$.

which coincides with equation (2.4) for $\lambda = 1$. Similarly, for $n = 4$ nodes $\{x_0 - h,\, x_0,\, x_0 + \lambda_1 h,\, x_0 + (\lambda_1 + \lambda_2)h\}$ the error is

$$\begin{aligned} \tau_4(x_0) = \; & -\lambda_1\,(\lambda_1 + \lambda_2)\,\frac{h^3}{24}\,u^{(IV)}(x_0) - \lambda_1\,(\lambda_1 + \lambda_2)\,\frac{h^3}{2\,c^2}\,u''(x_0) + \\ & + 3\,\lambda_1\,(\lambda_1 + \lambda_2)\,\frac{h^3}{8\,c^4}\,u(x_0) + O\left(h^4\,P_2(1/c^2)\right), \end{aligned} \tag{2.26}$$

which coincides with equation (2.5) for $\lambda_1 = 1$, $\lambda_2 = 1$.

Figure 2.9 shows the error as a function of c/h in the approximation of the first derivative with RBF-FD formula corresponding to three non-equispaced nodes. The results correspond to the numerical solution. The analytical results corresponding to equation (2.25) are not shown for clarity of the figure but they coincide with the numerical results. It can be observed that for large values of c/h the error of standard finite difference formulas is recovered. For smaller values of c/h the error decreases as $(h/c)^2$. Notice also that the optimal value of the shape parameter is independent of λ. This is to be expected since all the terms of order h^2 in equation (2.25) contain the factor λ, and therefore this factor disappears when equating the error to zero. Similarly, the optimal value of the shape parameter in the case of four non-equispaced nodes is also independent of λ since all the tems of order h^3 in equation (2.26) contain the factor $\lambda_1\,(\lambda_1 + \lambda_2)$.

In the case of the second derivative, the coefficients of the RBF-FD formula in the limit $c \gg h$ using three non-equispaced nodes $\{x_0 - h,\, x_0,\, x_0 + \lambda h\}$

are

$$\alpha_{-1} = \frac{2}{h^2(\lambda+1)} + \frac{(\lambda-3)\lambda}{(\lambda+1)c^2} + \frac{h^2(2\lambda(\lambda(2(\lambda-2)\lambda-1)-5)+1)}{4(\lambda+1)c^4} + O\left(\frac{h^4}{c^6}\right),$$

$$\alpha_0 = -\frac{2}{h^2\lambda} + \frac{-\lambda+4-\frac{1}{\lambda}}{c^2} - \frac{h^2\left((\lambda-3)\lambda\left(\lambda^2+1\right)+1\right)}{\lambda c^4} + O\left(\frac{h^4}{c^6}\right),$$

$$\alpha_1 = \frac{2}{h^2\left(\lambda^2+\lambda\right)} + \frac{\frac{1}{\lambda}-\frac{4}{\lambda+1}}{c^2} + \frac{h^2(\lambda(\lambda((\lambda-10)\lambda-2)-8)+4)}{4\lambda(\lambda+1)c^4} + O\left(\frac{h^4}{c^6}\right).$$

The corresponding formula for the approximation error is

$$\begin{aligned}\tau_3(x_0) = {} & \tfrac{\lambda-1}{3}\, h\, u'''(x_0) \,+\, (\lambda\,-\,1)\,\tfrac{h}{c^2}\, u'(x_0) + \\ & + \left[\lambda\,(\lambda\,-\,1)\,+\,1\right]\tfrac{h^2}{12}\, u^{(IV)}(x_0) + \\ & + \lambda\,\tfrac{h^2}{c^2}\, u''(x_0) \,+\, \left[\lambda\,(\lambda\,-\,5)\,+\,1\right]\tfrac{h^2}{4\,c^4}\, u(x_0) \,+\, O\left(h^3\, P_2(1/c^2)\right).\end{aligned} \tag{2.27}$$

This formula coincide with equation (2.11) when $\lambda = 1$. Notice that if $\lambda \neq 1$ the dependence of the error with h is only first order. Notice also that, to first order, the optimal value of the shape parameter, c^*, is independent of λ since all the terms of order h in equation (2.27) contain the same factor $(\lambda - 1)$.

Similar formulas can be derived for approximating the first and second derivatives with more nodes. For instance, the 4-node RBF-FD approximation to the second derivative using nodes $\{x_0{-}h,\, x_0,\, x_0{+}\lambda_1 h,\, x_0{+}(\lambda_1{+}\lambda_2)h\}$ is

$$\tau_4(x_0) \,=\, \left[\lambda_2 \,-\, \lambda_1\,(\lambda_1 \,+\, \lambda_2 \,-\, 2)\right]\left[\tfrac{h^2}{12}u^{(IV)}(x_0) \,+\, \tfrac{h^2}{c^2}\, u''(x_0) \,-\, \tfrac{3\,h^2}{4\,c^4}\, u(x_0)\right],$$

where the terms neglected are of order $O\left(h^3\, P_2(1/c^2)\right)$. As in previous cases the value of c^* is independent of the location of the nodes and of the local distance h. In 1D this result is general.

However, in 2D the value of c^* to leading order depends on the location of the nodes in the stencil but not on the nodal distance h. Consider for instance the equispaced 5-node stencil approximation of the Laplacian, in which we move the location of one node. Thus, the coordinates of the five nodes are $\{(x_0,\, y_0),\, (x_0,\, y_0 + \lambda\, h),\, (x_0 + h,\, y_0),\, (x_0,\, y_0 - h),\, (x_0 - h,\, y_0)\}$. Figure 2.10 shows the value of c^* as a function of λ corresponding to the Laplacian of function (2.18). For $\lambda = 1$, the value of c^* for equispaced nodes is recovered ($c^* \approx 0.5658$). For other values of λ the value of c^* varies continuously. Notice that when $c^* \gg h$, the value of c^* is independent of h. For $c^* = O(h)$

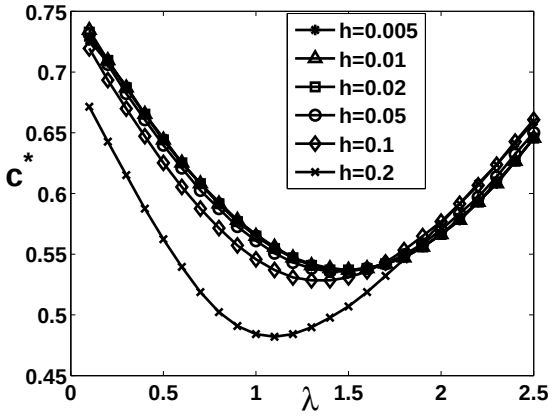


Figure 2.10: Optimal value of the shape parameter in the approximation of the Laplacian of function (2.18) at $\mathbf{x}_0 = (0, 0)$ with $n = 5$ non-equispaced RBF-FD formula.

there are corrections of higher order that come into play. In the case of fully arbitrary nodes the analysis is more complex but can be carried out in the same manner described in Section 2.4.

To understand the relationship between standard finite differences and RBF-FD formulas, consider the function value at a node $\mathbf{x}_i$ expressed by a Taylor expansion

$$u(\mathbf{x}_i) \,=\, u(\mathbf{x}_0) \,+\, \nabla u(\mathbf{x}_1) \,\cdot\, \bar{\mathbf{x}}_i \,+\, \frac{1}{2}\,\nabla^2 u(\mathbf{x}_0) \,:\, (\bar{\mathbf{x}}_i \,\cdot\, \bar{\mathbf{x}}_i^T) \,+\, e_i\,,$$

where $\bar{\mathbf{x}}_i \,=\, \mathbf{x}_i \,-\, \mathbf{x}_0$, and e_i is the error in the expansion. Here, we have denoted the matrix scalar product by ':'. A linear combination with coefficients $\{\alpha_i\}_{i=1}^n$ equals

$$\begin{aligned}\sum_{i=1}^{n} \alpha_i\, u(\mathbf{x}_i) \quad &=\quad u(\mathbf{x}_0)\,\left(\sum_{i=1}^{n} \alpha_i\right) \,+\, \nabla u(\mathbf{x}_0) \,\cdot\, \left(\sum_{i=2}^{n} \alpha_i\, \bar{\mathbf{x}}_i\right) \,+ \\ &\qquad +\frac{1}{2}\,\nabla^2 u(\mathbf{x}_0) \,:\, \left(\sum_{i=2}^{n} \alpha_i\,(\bar{\mathbf{x}}_i \,\cdot\, \bar{\mathbf{x}}_i^T)\right) \,+\, \left(\sum_{i=2}^{n} \alpha_i\, e_i\right).\end{aligned}$$

This FD formula approximates the Laplacian to first order exactly for constant, linear and quadratic functions, provided that the coefficients satisfy the following conditions:

$$\sum_{i=1}^{n} \alpha_i \,=\, 0, \quad \sum_{i=2}^{n} \alpha_i\, \bar{\mathbf{x}}_i \,=\, 0, \quad \sum_{i=2}^{n} (\bar{\mathbf{x}}_i \,\cdot\, \bar{\mathbf{x}}_i^T)\, \alpha_i \,=\, 2\, I\,. \tag{2.28}$$

These are a total of 6 conditions which have to be satisfied for the approximation to be *consistent* [94]. In matrix form, with $\bar{\mathbf{x}}_i = (\bar{x}_i, \bar{y}_i)$, we can write (2.28) as

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \bar{x}_2 & \bar{x}_3 & \dots & \bar{x}_n \\ 0 & \bar{y}_2 & \bar{y}_3 & \dots & \bar{y}_n \\ 0 & \bar{x}_2\bar{y}_2 & \bar{x}_3\bar{y}_3 & \dots & \bar{x}_n\bar{y}_n \\ 0 & \bar{x}_2^2 & \bar{x}_3^2 & \dots & \bar{x}_n^2 \\ 0 & \bar{y}_2^2 & \bar{y}_3^2 & \dots & \bar{y}_n^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 2 \end{bmatrix}.$$

Thus, if six nodes are used in the stencil and the matrix has full rank, there is a unique set of coefficients α_i that satisfy the constraints (2.28). If $n < 6$ there is no solution and if $n > 6$ there are infinitely many solutions. In this case, a unique set of coefficients can be derived, for instance, by the generalized finite difference method (GFDM) [73] or by moving least squares methods [62]. If Taylor series is carried out until next order and the FD formula is required to be exact also for cubic functions, then four additional constraints have to be satisfied (corresponding to the coefficients of $u^{(3,0)}$, $u^{(0,3)}$, $u^{(2,1)}$, $u^{(1,2)}$). Thus, a unique solution will exist for $n = 10$. In general, if the system is full rank a unique solution of order p exists for $n = (p+2)(p+3)/2$ (so called triangle numbers).

With RBF-FD this limitation does not exist. In fact, adding a new node to an existing stencil also adds the corresponding RBF to the basis of the functional space. Therefore the matrix associated to system (1.18) is always square and, provided it is of full rank, it has a unique solution. For values of n for which the standard finite difference formulation has a unique solution, the coefficients of RBF-FD in the limit $c \to \infty$ are identical to the coefficients of standard finite differences. Thus, the order of RBF-FD formulas coincide with the order of the corresponding finite difference formulas (order 1 for $6 \leq n \leq 9$, order 2 for $10 \leq n \leq 14$, order 3 for $15 \leq n \leq 20$, ...).

Consider, for instance, the case of 6 nodes. Following the same procedure described in Section 2.4, the error of approximation in the limit $c \gg h$ can be expressed as

$$\begin{aligned} \tau_6(\mathbf{x}_0) = \; & h \left[A_{0,0}\, u^{(3,0)}(\mathbf{x}_0) + A_{0,1}\, u^{(2,1)}(\mathbf{x}_0) + \right. \\ & \left. + A_{0,2}\, u^{(1,2)}(\mathbf{x}_0) + A_{0,3}\, u^{(0,3)}(\mathbf{x}_0) \right] + \\ & + \tfrac{h}{c^2} \left[A_{1,0}, u^{(1,0)}(\mathbf{x}_0) + A_{1,1}\, u^{(0,1)}(\mathbf{x}_0) \right] + O\left(h^2\, P_2(1/c^2)\right), \end{aligned} \tag{2.29}$$

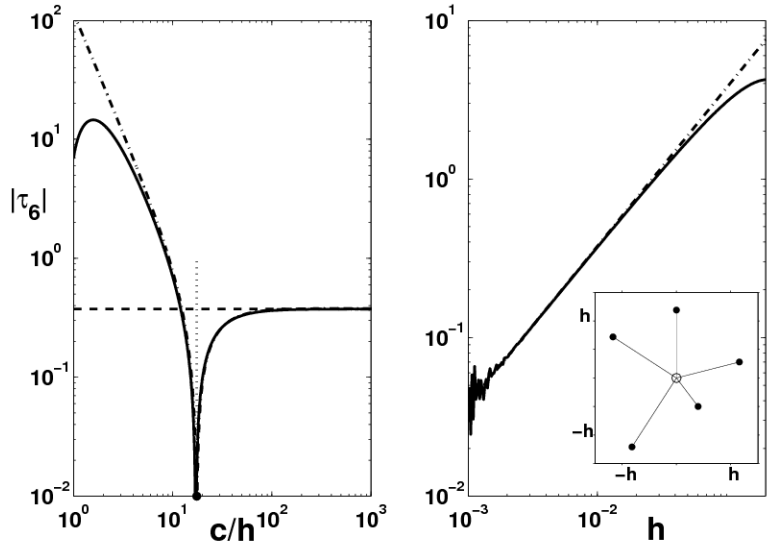


Figure 2.11: Approximation error for Laplacian of function (2.18) at $\mathbf{x}_0 = (0,0)$ with $n = 6$ non-equispaced RBF-FD formula. $\mathbf{x}$ $[(0,0), (-1.17h, 0.72h), (-0.82h, -1.21h), (0.4h, -0.5h), (1.16h, 0.28h), (0, 1.19h)]$. Left: dependence with c/h ($h = 0.01$). Right: dependence with h ($c = 2$). Solid line; numerical results. Dot-dashed line: equation (2.29). Dashed line: finite differences. Dotted line: optimal value c^*.

where the coefficients $A_{i,j}$ are constants which can be computed for a given node distribution. Notice that the error is of order h, like with standard finite differences. The coefficients $A_{0,i}$ satisfy the compatibility constraints (2.28) and therefore coincide with the coefficients of the standard 6-node FD formula.

Figure 2.11 compares the analytical approximation of the error given by equation (2.29) (dot-dashed line) with the actual numerical error (solid line) for a specific node distribution shown in the right side of the figure. Similarly to what was observed for the case of equispaced nodes (Figure 2.7) there is an optimal value of the shape parameter ($c^* \approx 0.1754$) for which the error becomes zero. This value is shown by a dotted line in the left side of the figure. To the left of that minimum the error decreases as c^{-2} and to the right of that minimum it approaches the error corresponding to standard finite differences. As before, the value of c^* is simply obtained by equating to zero equation (2.29).

The right side of Figure 2.11 compares the dependence of the error with

h given by equation (2.29) to the numerically computed dependence. To obtain the numerical dependence with h we use the same distribution shown in the inset of the figure but vary its scale with h. Both results are in good agreement until the onset of ill-conditioning and show that the error reduction is $O(h)$.

Similar results can be obtained for any value of n. In fact, it is possible to derive a general formula for the error in approximating the Laplacian with n non-equispaced nodes in the limit $c \gg h$. This formula is the analogous of equation (2.17) for unstructured grids;

$$\tau_n(\mathbf{x}_0) \;=\; h^s \, P_{(s-k)/2+1}(1/c^2), \tag{2.30}$$

where $k = 0$ if p even and $k = 1$ if p odd, and

$$\frac{(s+2)(s+3)}{2} \leq n \leq \frac{(s+3)(s+4)}{2}.$$

The polynomial takes the form

$$P_{(s-k)/2+1}(1/c^2) = \sum_{m=0}^{(s-k)/2+1} \; \sum_{r=0}^{s+2(1-m)} \frac{A_{m,r}}{c^{2\,m}} \, u^{(s+2(1-m)-r\,,r)}(\mathbf{x}_0).$$

Equation (2.30) has the same form than equation (2.19) and, therefore, the procedure described in Section 2.5 for computing the optimal value of the shape parameter is also applicable to the case of non-equispaced nodes.

2.7 Conclusions

We have derived series solutions in powers of the shape parameter c, and nodal distance h, for the local truncation error in approximating differential operators with RBF-FD formulas. The main conclusions of this chapter are the following:

- RBF-FD formulas approach conventional finite difference formulas in the limit of infinitely flat basis functions ($c \gg h$).
- For each formula, there is a range of values of the shape parameter for which RBF-FD formulas are significantly more accurate than the corresponding conventional finite difference formulas.
- In the case of equispaced nodes, Ding *et.al.* [20] concluded that the error dependence with c and h is $\tau_n \;=\; O\left((h/c)^k\right)$, with $k \approx 1.9$ for

$6 \leq n \leq 9$, $k \approx 3.6$ for $9 < n \leq 27$, $k \approx 4.9$ for $27 < n \leq 34$. However, we find that $\tau_n = O\left(h^s P_k(1/c^2)\right)$, where s depends on the number of nodes n and $P_k(1/c^2)$ is a polynomial or order k.

- For equispaced nodes; $s = 2$ for $5 \leq n \leq 12$, $s = 4$ for $13 \leq n \leq 28$, and $s = 6$ for $29 \leq n \leq 52$, ...
- For non-equispaced nodes; $s = 1$ for $6 \leq n \leq 9$, $s = 2$ for $10 \leq n \leq 14$, $s = 3$ for $15 \leq n \leq 20$, ...
- There are specific values of n for which the error is significantly smaller than the error for $n - 1$. These values should be used in practical applications. For equispaced nodes; $n = (s-1)^2 + 4$, where the order s is any even number. For non-equispaced nodes; $n = (s+2)(s+3)/2$ where the order s is any integer.
- For each RBF-FD formula there is an optimal value of the shape parameter c^* for which the error is minimum. To leading order, this value is independent of h and only depends on the value of the function and its derivatives at the node.

Chapter 3

Optimal constant shape parameter for multiquadric based RBF-FD method

3.1 Outline

The development of RBF-FD error formulas carried out in the previous chapter reveals the exact dependence of the optimal shape parameter on the value of the function and its derivatives at the node. Based on these formulas, we present in this chapter a novel technique which allow us to select the value of the shape parameter in order to minimize the error of the approximation to the solution of a PDE. We show that the accuracy of the solution can be improved one or two orders of magnitude with respect to finite differences if one selects the right value of the shape parameter. The technique is developed for MQ as RBFs, but it is readily applicable to any other infinitely smooth RBF such as gaussians (Appendix A).

The chapter is organized as follows. In Section 3.2, we describe how to compute the optimal constant shape parameter using the RBF-FD method. In Section 3.3, we describe in detail the resulting numerical algorithm. Section 3.4 includes several examples in 1D and 2D using both structured and unstructured nodes. Finally, in Section 3.5 we summarize the main results of this chapter.

3.2 Optimal Constant Shape Parameter

Consider the Dirichlet problem in a bounded domain $\Omega \subset \mathbb{R}^d$

$$\begin{cases} \mathcal{L}u(\mathbf{x}) &= f(\mathbf{x}), & \text{in } \Omega \\ u(\mathbf{x}) &= g(\mathbf{x}), & \text{on } \partial\Omega \end{cases} \tag{3.1}$$

where $\mathcal{L}$ is a differential operator and $f(\mathbf{x})$ and $g(\mathbf{x})$ are real functions. Let the domain Ω be discretized using N scattered nodes (N_I interior nodes and $N - N_I$ boundary nodes). Following the procedure described in Section 1.4, we approximate $\mathcal{L}u$ at a node $\mathbf{x}_i$ by a linear combination of the values of the unknown function u at n scattered nodes surrounding $\mathbf{x}_i$, which constitute its stencil. Thus, equation (3.1) can be written at $\mathbf{x}_i$ as

$$\sum_{j=1}^{n} \alpha_{ij}(c) u(\mathbf{x}_j) = f(\mathbf{x}_i) + \tau_n(\mathbf{x}_i\,;\, c), \qquad 1 \le i \le N_I, \tag{3.2}$$

where $\tau_n(\mathbf{x}_i\,;\, c)$ is the local truncation error resulting from the approximation. In matrix form, these equations are written as

$$A(c)\mathbf{u} = \mathbf{f} + \boldsymbol{\tau}(c)\ , \tag{3.3}$$

where $A(c)$ is a $N_I \times N_I$ sparse matrix whose entries are the weighting coefficients $\alpha_{ij}(c)$, $\mathbf{u}$ is the row vector of exact solutions at the interior nodes, and $\boldsymbol{\tau}(c)$ is a row vector formed by the local truncation errors $\tau_n(\mathbf{x}_i\,;\, c)$ at these nodes.

The RBF-FD approximation $\hat{\mathbf{u}}$ to the exact solution $\mathbf{u}$ is

$$\hat{\mathbf{u}}(c) = A^{-1}(c)\mathbf{f}\ , \tag{3.4}$$

so that the global RBF-FD error is given by

$$\mathbf{E}(c) \equiv \mathbf{u} - \hat{\mathbf{u}}(c) = A^{-1}(c)\,\boldsymbol{\tau}(c). \tag{3.5}$$

Taking the norm of the error gives

$$\|\mathbf{E}(c)\| = \left\|A^{-1}(c)\,\boldsymbol{\tau}(c)\right\| \le \left\|A^{-1}(c)\right\|\,\|\boldsymbol{\tau}(c)\|\,. \tag{3.6}$$

From the results of Chapter 2 and Appendix A we already know that $\|\boldsymbol{\tau}(c)\| = O\left(h^n P_k(1/c^2)\right)$. The global RBF-FD error $\|\mathbf{E}(c)\|$ preserves the convergence of $\|\boldsymbol{\tau}(c)\|$ and goes to zero at least as fast if $\|A^{-1}(c)\|$ is bounded by a constant K independent of h and c as $h/c \to 0$, i.e.

$$\|\mathbf{E}(c)\| \le K\,\|\boldsymbol{\tau}(c)\|\,. \tag{3.7}$$

Since the RBF-FD method is equivalent to FD in the limit $c \to \infty$, $\|A^{-1}(c)\|$ is guaranteed to be bounded in that limit. In the following sections, we will see through numerical examples that this result holds for $c \gg h$ in all the cases considered.

At this point, the problem of finding the *optimal* shape parameter can be stated as the problem of finding the value of the shape parameter c which minimizes (3.6) in a certain norm. Thus, we define the *optimal constant shape parameter* as the value c^* such that

$$\|\mathbf{E}(c^*)\|_\infty = \min_c \|\mathbf{E}(c)\|_\infty = \min_c \left\|A^{-1}(c)\,\boldsymbol{\tau}(c)\right\|. \tag{3.8}$$

The local truncation error $\boldsymbol{\tau}(c)$ derived in Chapter 2 are written as series expansions in powers of h (the inter-nodal distance), which are valid for $c \gg h$. The coefficients of these formulas depend on c, h, and on the value of the exact solution and its derivatives. Therefore, it is apparent that in real problems the value c^* can not be computed directly from (3.8) because the exact solution and its derivatives are not known.

However, as we will show later, it can be estimated using an approximation $\tilde{\mathbf{u}}$ to the exact solution $\mathbf{u}$ without losing accuracy to leading order. In this work, we use finite differences to compute $\tilde{\mathbf{u}}$ with both structured or unstructured grids.

Therefore, we look for an approximate value $\tilde{c}^*$ to the optimal shape parameter c^* such that

$$\left\|\tilde{\mathbf{E}}(\tilde{c}^*)\right\|_\infty = \min_c \left\|A^{-1}(c)\,\tilde{\boldsymbol{\tau}}(c)\right\|_\infty , \tag{3.9}$$

where $\tilde{\boldsymbol{\tau}}(c)$ is the estimated local truncation error.

3.3 Numerical algorithm

For a given problem (3.1), the numerical algorithm which implements the optimal constant shape parameter is as follows:

1. For each interior node $\mathbf{x}_i$ determine a stencil of n surrounding nodes.
2. Use finite differences to compute an approximate solution $\tilde{\mathbf{u}}(\mathbf{x}_i)$.
3. Find the value $\tilde{c}^*$ which minimizes $||A^{-1}(c)\,\tilde{\boldsymbol{\tau}}(c)||_\infty$. At each iteration:
   - Use (1.18) to compute numerically the RBF-FD coefficients $\alpha_{ij}(c)$, and therefore matrix $A(c)$.

- Use the finite difference solution $\tilde{\mathbf{u}}(\mathbf{x}_i)$ computed in step 2 and the analytical formulas derived in Chapter 2 to estimate the local approximation error $\tau_n(\mathbf{x}_i\,;\,c)$ at each node, and therefore $\tilde{\boldsymbol{\tau}}(c) = [\tilde{\tau}(\mathbf{x}_1\,;\,c), \tilde{\tau}(\mathbf{x}_2\,;\,c), \ldots, \tilde{\tau}(\mathbf{x}_{N_I}\,;\,c)]^T$.

4. Compute the optimal RBF-FD approximate solution $\hat{\mathbf{u}}(\tilde{c}^*) = A^{-1}(\tilde{c}^*)\,\mathbf{f}$.

In step 3 we have used MATLAB command `fminsearch` which finds the minimum of a nonlinear function using the Nelder-Mead Simplex method. From the point of view of computational cost, the algorithm requires the solution of two $N_I \times N_I$ sparse linear systems (steps 2 and 4), and the solution of N_I dense systems of $n \times n$ unknowns at each iteration of step 3. The typical number of iterations required is 15.

A possibility to try to improve the accuracy of $\tilde{c}^*$ is to use an iterative procedure: once an approximate value of $\tilde{c}^*$ is computed in step (4) and therefore a better approximation $\hat{\mathbf{u}}(\tilde{c}^*)$ to the exact solution, go back to step (3) to compute a refined value of $\tilde{c}^*$.

3.4 Example problems

In this section we will apply the numerical algorithm just described to the solution of some example problems in 1D and 2D, using both structured and non structured nodes. We will also apply the technique to several problems that have been solved in the past with the RBF-FD method. We will show that using the optimal constant value of the shape parameter leads to significant improvements in accuracy.

3.4.1 One dimensional boundary value problem

Consider the following problem

$$\begin{cases} u_{xx} &= f(x), \qquad 0 < x < 1 \\ u(0) &= 1, \qquad u(1) = 1 + \dfrac{\sqrt{2}}{2} \end{cases} \tag{3.10}$$

where $f(x)$ is computed from the known solution $u(x) = 1 - \sin\left(\dfrac{5\pi}{4}x\right)$.

3.4.1.1 Structured nodes

Let us discretize the domain in (3.10) using N structured nodes and let's use a three nodes $\{x-h, x, x+h\}$ central difference scheme to approximate the

Table 3.1: Optimal shape parameters and the corresponding errors for problem (3.10) with structured nodes.

	$N = 21$	$N = 41$	$N = 61$	$N = 81$
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$4.181 \cdot 10^{-3}$	$1.044 \cdot 10^{-3}$	$4.638 \cdot 10^{-4}$	$2.609 \cdot 10^{-4}$
c^*	0.8589	0.8616	0.8621	0.8623
$\|\|\mathbf{E}\,(c^*)\,\|\|_\infty$	$6.694 \cdot 10^{-5}$	$1.672 \cdot 10^{-5}$	$7.425 \cdot 10^{-6}$	$4.182 \cdot 10^{-6}$
$\tilde{c}^*$	0.8628	0.8626	0.8625	0.8625
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$7.763 \cdot 10^{-5}$	$1.735 \cdot 10^{-5}$	$7.541 \cdot 10^{-6}$	$4.216 \cdot 10^{-6}$
$\|c^* - \tilde{c}^*\|$	0.0039	0.0010	0.0004	0.0002

second derivative. The resulting local truncation error (2.11) is estimated as

$$\tilde{\tau}_3(x;c) = \frac{h^2}{12}\, f''(x) \,+\, \frac{h^2}{c^2}\, f(x) \,-\, \frac{3\,h^2}{4\,c^4}\, \tilde{u}(x) \,+\, O\left(h^4\, P_3(1/c^2)\right), \quad (3.11)$$

where $\tilde{u}(x)$ is a finite difference approximation to the exact solution $u(x)$, so $\tilde{u}(x) = u(x) + O(h^2)$, and $u''(x)$ and $u^{(IV)}$ are computed exactly from $f(x)$. Notice that the accuracy of the estimated local error (3.11) is $O(h^4)$ as the leading error of the analytical formula (2.11).

Figure 3.1 shows with solid lines the infinity norm of the global RBF-FD error, $||\mathbf{E}(c)||_\infty = ||\mathbf{u} - \hat{\mathbf{u}}(c)||_\infty$, as a function of the shape parameter c for different number of nodes N. In all cases, the estimated global error $||\tilde{\mathbf{E}}(c)||_\infty = \|A^{-1}(c)\tilde{\boldsymbol{\tau}}_3(c)\|_\infty$, shown with dot-dashed lines, reproduces closely the global error in the limit $c \gg h$. This allows us to estimate c^* accurately. Notice that for large values of c, the RBF-FD error approaches the standard finite-difference error, depicted in Fig. 3.1 with dashed lines. We observe that there is a range of shape parameters around c^* for which the RBF-FD solution is significantly more accurate than the finite difference solution. Notice also that in the case $N = 81$ ill conditioning appears for large values of c.

From top to bottom, Table 3.1 shows the infinite norm of the finite difference solution error $\|\mathbf{E}_{FD}\|_\infty$, the exact optimal shape parameter c^* and its corresponding error $||\mathbf{E}\,(c^*)\,||_\infty$, the estimated optimal shape parameter $\tilde{c}^*$ and its corresponding error $||\mathbf{E}(\tilde{c}^*)||_\infty$, and the difference between the exact and the estimated optimal shape parameter, which is $O(h^2)$. Notice that, to leading order, the value of the optimal shape parameter is independent of h (independent of N). Also notice that there is an improvement of approximately two orders of magnitude between the finite difference solution and the optimal RBF-FD solution.

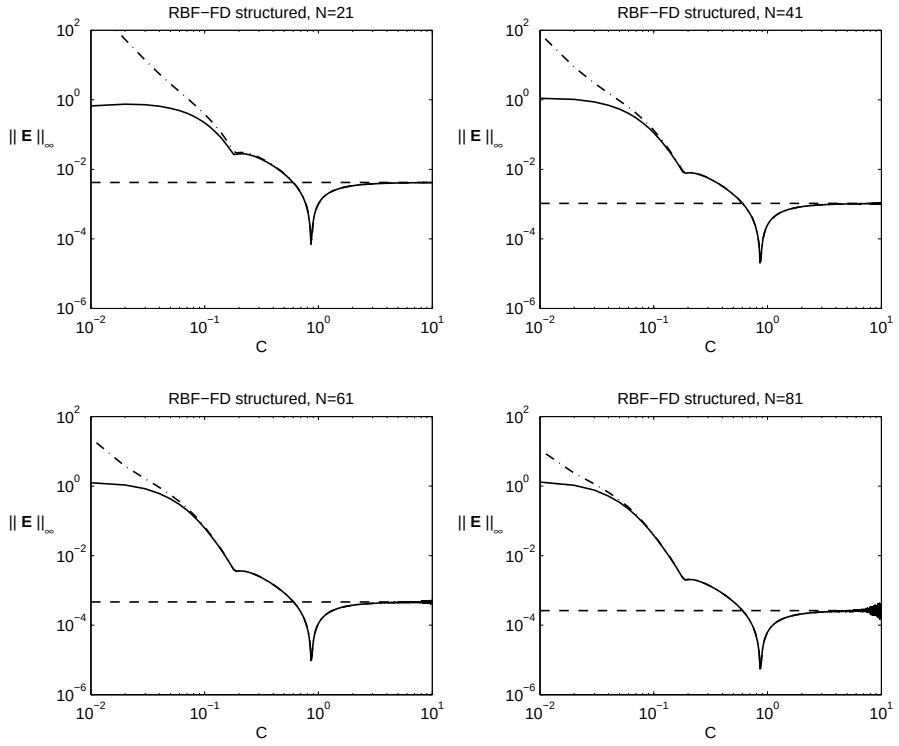


Figure 3.1: Infinite norm of the errors of problem (3.10) as function of c, using $N = 21, 41, 61$, and 81 structured nodes. Solid lines: global RBF-FD error (3.5). Dot-dashed lines: estimated error (3.11). Dashed lines: finite difference error.

Table 3.2: Same as Table 3.1 but for unstructured nodes.

	$N = 21$	$N = 41$	$N = 61$	$N = 81$
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$4.928 \cdot 10^{-3}$	$1.327 \cdot 10^{-3}$	$3.818 \cdot 10^{-4}$	$3.377 \cdot 10^{-4}$
c^*	0.7795	0.7944	1.0063	0.8067
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$1.178 \cdot 10^{-3}$	$1.945 \cdot 10^{-4}$	$4.551 \cdot 10^{-5}$	$2.580 \cdot 10^{-5}$
$\tilde{c}^*$	0.7858	0.7959	1.0069	0.8071
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$1.207 \cdot 10^{-3}$	$1.981 \cdot 10^{-4}$	$4.586 \cdot 10^{-5}$	$2.595 \cdot 10^{-5}$
$\|c^* - \tilde{c}^*\|$	0.0063	0.0015	0.0006	0.0004

3.4.1.2 Unstructured nodes

In the case that the domain is discretized with unequally spaced nodes, the local truncation error using a three nodes $\{x-h, x, x+\lambda h\}$) central difference scheme is

$$\begin{aligned}\tau_3(x;c) &= \frac{\lambda - 1}{3} h\, u'''(x) + (\lambda - 1)\, \frac{h}{c^2}\, u'(x) \\ &+ [\lambda\,(\lambda - 1) + 1]\, \frac{h^2}{12}\, u^{(IV)}(x) + \lambda \frac{h^2}{c^2}\, u''(x) \\ &+ [\lambda\,(\lambda - 5) + 1]\, \frac{h^2}{4\,c^4}\, u(x) + O\left(h^3\, P_2(1/c^2)\right). \quad (3.12)\end{aligned}$$

In this case, the local approximation error is only of order $O(h)$ so we also include terms of order $O(h^2)$ in the formula. The resulting approximation to the local error is of order $O(h^3)$ while in the case of structured nodes it was of order $O(h^4)$. To compute the estimated error $\tilde{\tau}_3(x;c)$ the derivatives of order greater or equal to two appearing in equation (3.12) are computed exactly from the derivatives of f. The values of u and u' are approximated to first order using finite differences.

Figure 3.2 shows the corresponding infinite norm of the error $||\mathbf{E}(c)||_\infty$ as function of c for different number of nodes N (we use Halton nodes [49] here). In this case, the error estimation (dot-dashed lines) is as accurate as in the case of structured nodes, but the minimum error corresponding to $\tilde{c}^*$ is less pronounced.

Table 3.2 shows the same information as Table 3.1 but for unstructured nodes. As before, the optimal shape parameter is estimated accurately ($c^* - \tilde{c}^* = O(h^2)$), but the improvements in accuracy with respect to finite differences are less significant (approximately one order of magnitude).

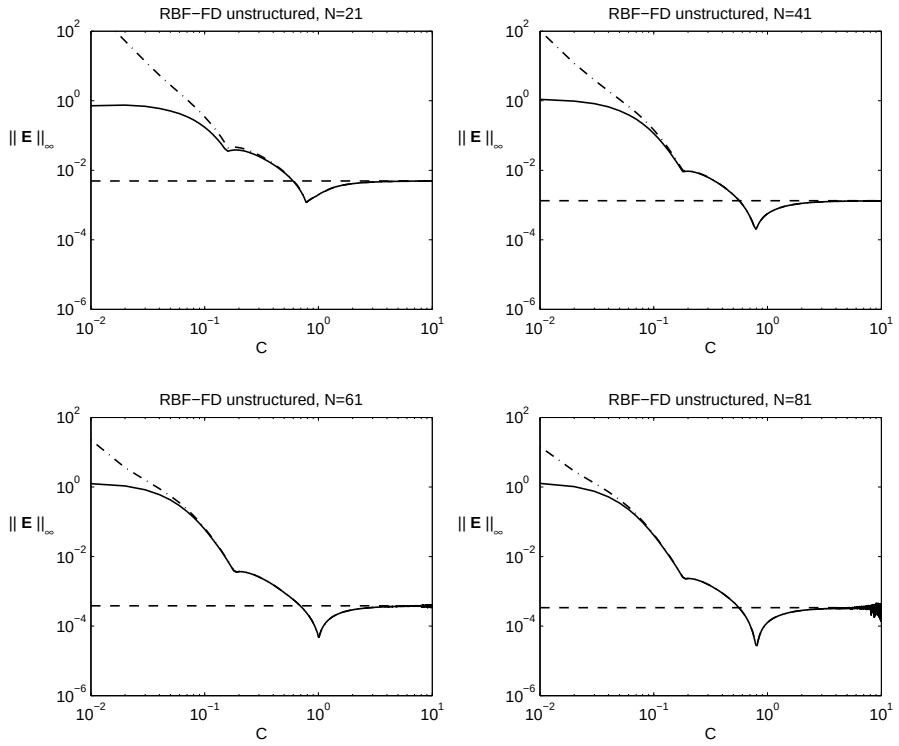


Figure 3.2: Same as Fig. 3.1 but for unstructured nodes.

3.4.2 Steady convection-diffusion problem

Consider the problem

$$\begin{cases} u_x - u_{xx} = \pi^2 \sin(\pi x) + \pi \cos(\pi x), & 0 < x < 1, \\ u(0) = 0, \qquad u(1) = 1 \end{cases} \tag{3.13}$$

whose exact solution is $u(x) = \sin(\pi x) + \dfrac{e^x - 1}{e - 1}$. This problem was proposed and solved in [11].

3.4.2.1 Structured nodes

The local approximation error to the convection-diffusion differential operator with the RBF-FD formula using three structured nodes is

$$\begin{aligned} \tau_3(x;c) &= \frac{h^2}{12}\left(2\,u'''(x) - u^{(IV)}(x)\right) + \frac{h^2}{2c^2}\left(u'(x) - 2\,u''(x)\right) \\ &+ \frac{3\,h^2}{4\,c^4}\,u(x) + O\left(h^4\,P_3(1/c^2)\right). \end{aligned} \tag{3.14}$$

In this formula u is approximated using a second order central difference scheme, and u' is approximated from $\tilde{u}$ using the corresponding second order central difference scheme. Higher derivatives are approximated to second order through the recursion $\tilde{u}^{(k+1)} = \tilde{u}^{(k)} - f^{(k-1)}$ for $k \geq 1$.

Figure 3.3 shows with solid lines the corresponding infinite norm of the error $||\mathbf{E}(c)||_\infty$ as a function of c for different number of nodes N. As in the previous cases, the error estimated with the analytical formulas is in close agreement with the actual error, and therefore, the optimal shape parameter can be estimated accurately.

Table 3.3 shows the same information as Table 3.1 but for problem (3.13). In reference [11] the optimal shape parameter was computed by trial and error ($c^* = 1/\varepsilon^* = 1/0.9 = 1.11$). The authors included a constant term in the RBF interpolant in order to impose the condition that the RBF-FD formulas are exact for constants (see equation (1.25) in Section 1.4). The corresponding results were presented in Table IV of [11], and are reproduced here in the fourth and fifth rows of Table 3.3.

3.4.2.2 Unstructured nodes

In the case that the domain is discretized with unequally spaced nodes, the local RBF-FD approximation error using a three nodes $\{x - h, x, x + \lambda h\}$

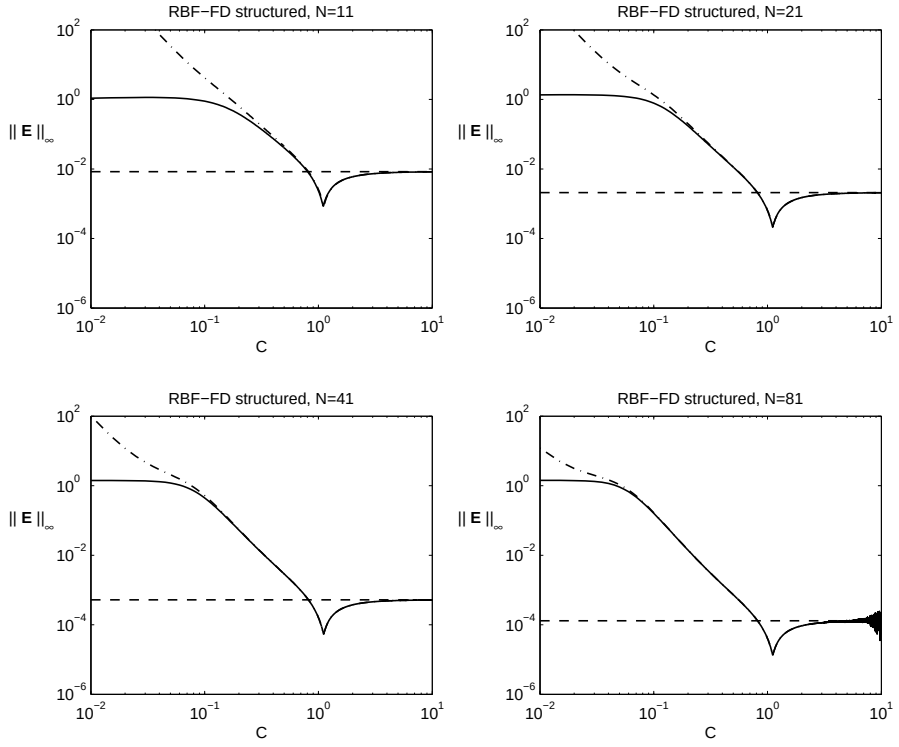


Figure 3.3: Infinite norm of the errors of problem (3.13) as function of c, using $N = 11, 21, 41$, and 81 structured nodes. Solid lines: global RBF-FD error (3.5). Dot-dashed lines: estimated error (3.14). Dashed lines: finite difference error.

Table 3.3: Optimal shape parameters and the corresponding errors for problem (3.13) with structured nodes.

	$N = 11$	$N = 21$	$N = 41$	$N = 81$
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$8.337 \cdot 10^{-3}$	$2.088 \cdot 10^{-3}$	$5.220 \cdot 10^{-4}$	$1.305 \cdot 10^{-4}$
c^*	1.1031	1.1101	1.1116	1.1121
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$8.318 \cdot 10^{-4}$	$2.090 \cdot 10^{-4}$	$5.241 \cdot 10^{-5}$	$1.311 \cdot 10^{-5}$
$c^*_{[11]}$	1/0.9	1/0.9	1/0.9	1/0.9
$\|\|\mathbf{E}(c^*_{[11]})\|\|_\infty$	$1.43 \cdot 10^{-3}$	$3.83 \cdot 10^{-4}$	$9.75 \cdot 10^{-5}$	$2.45 \cdot 10^{-5}$
$\tilde{c}^*$	1.1139	1.1127	1.1123	1.1123
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$9.245 \cdot 10^{-4}$	$2.161 \cdot 10^{-4}$	$5.282 \cdot 10^{-5}$	$1.314 \cdot 10^{-5}$
$\|c^* - \tilde{c}^*\|$	0.0108	0.0026	0.0007	0.0002

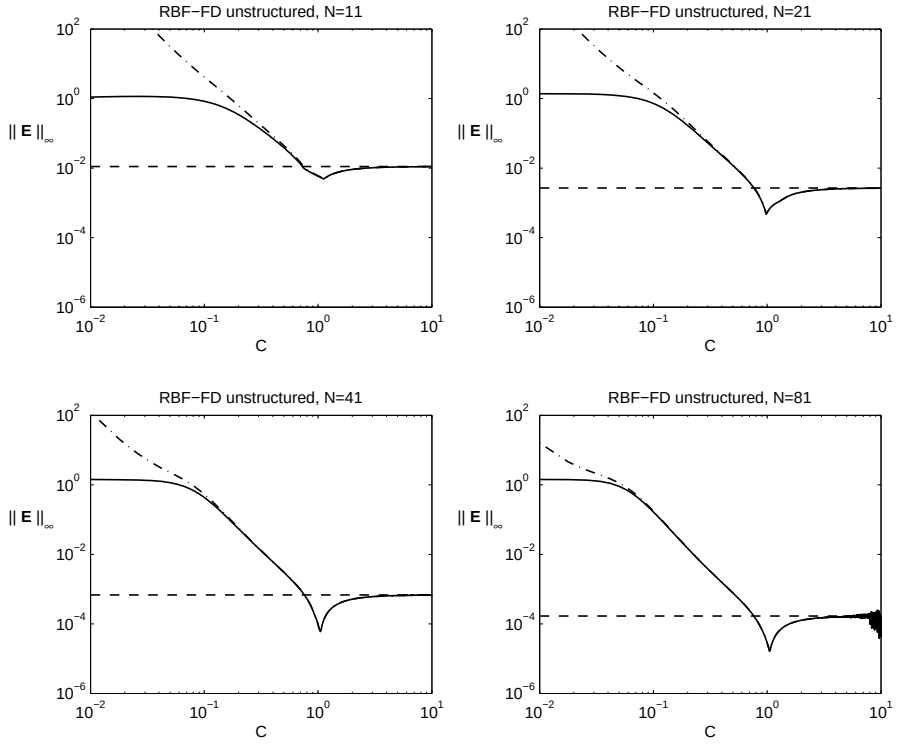


Figure 3.4: Same as Fig. 3.3 but for unstructured nodes.

central difference scheme for the convection-diffusion operator is

$$\begin{aligned}\tau_3(x;c) &= \frac{(1-\lambda)}{3}\,h\,u'''(x) + (1-\lambda)\,\frac{h}{c^2}\,u'(x)\\ &+ \frac{h^2}{12}\left[2\,\lambda\,u'''(x) - [\lambda\,(\lambda-1)+1]\,u^{(IV)}(x)\right]\\ &+ \frac{\lambda\,h^2}{2\,c^2}\,[\,u'(x) - 2\,u''(x)\,]\\ &- [\lambda\,(\lambda-5)+1]\,\frac{h^2}{4\,c^4}\,u(x) + O\left(h^3\,P_2(1/c^2)\right). \qquad (3.15)\end{aligned}$$

As was the case in (3.12), the local approximation error is only of order $O(h)$ so we include terms of order $O(h^2)$ in this formula too. The derivatives appearing in these formulas are approximated in the same way described in the previous section. The resulting approximation to the local error is also of order $O(h^3)$ while in the case of structured nodes it was of order $O(h^4)$.

Table 3.4: Same as Table 3.3 but for unstructured nodes.

	$N = 11$	$N = 21$	$N = 41$	$N = 81$
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$1.105 \cdot 10^{-2}$	$2.667 \cdot 10^{-3}$	$6.788 \cdot 10^{-4}$	$1.685 \cdot 10^{-4}$
c^*	1.1230	0.9773	1.0525	1.0536
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$4.863 \cdot 10^{-3}$	$4.690 \cdot 10^{-4}$	$5.891 \cdot 10^{-5}$	$1.578 \cdot 10^{-5}$
$\tilde{c}^*$	1.1409	0.9818	1.0536	1.0539
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$5.060 \cdot 10^{-3}$	$4.832 \cdot 10^{-4}$	$5.979 \cdot 10^{-5}$	$1.584 \cdot 10^{-5}$
$\|c^* - \tilde{c}^*\|$	0.0179	0.0045	0.0011	0.0003

Figure 3.4 shows with solid lines the corresponding infinite norm of the error $||\mathbf{E}(c)||_\infty$ as function of c for different number of nodes N (Halton nodes). The error estimation (dot-dashed lines) is as accurate as in the case of structured nodes.

Table 3.4 summarizes these results. Notice that in this case the optimal shape parameter c^* is more dependent on h. This is due to the fact that the formula for the error is only order h and therefore, c^* is independent of h only for large values of N.

3.4.3 Two dimensional boundary value problem

Consider now the two dimensional Poisson problem

$$\begin{cases} \Delta u(\mathbf{x}) &= f(\mathbf{x}), & \text{in } \Omega = (0,1) \times (0,1) \\ u(\mathbf{x}) &= g(\mathbf{x}), & \text{on } \partial\Omega \end{cases} \tag{3.16}$$

where $f(\mathbf{x})$ and $g(\mathbf{x})$ are obtained from the exact solution

$$u(x,y) = \exp\left[-\left(x-\frac{1}{4}\right)^2 - \left(y-\frac{1}{2}\right)^2\right] \cos(2\,\pi\,y)\,\sin(\pi\,x)\ . \tag{3.17}$$

This problem has been used by Wright and Fornberg [109] to test the performance of the RBF-FD and RBF-HFD methods.

3.4.3.1 Structured nodes

Suppose the domain is discretized in $N \times N$ structured nodes. Using a five nodes $\{(x,y),(x-h,y),(x+h,y),(x,y-h),(x,y+h)\}$ central difference scheme, the local truncation error is given by equation (2.14). In

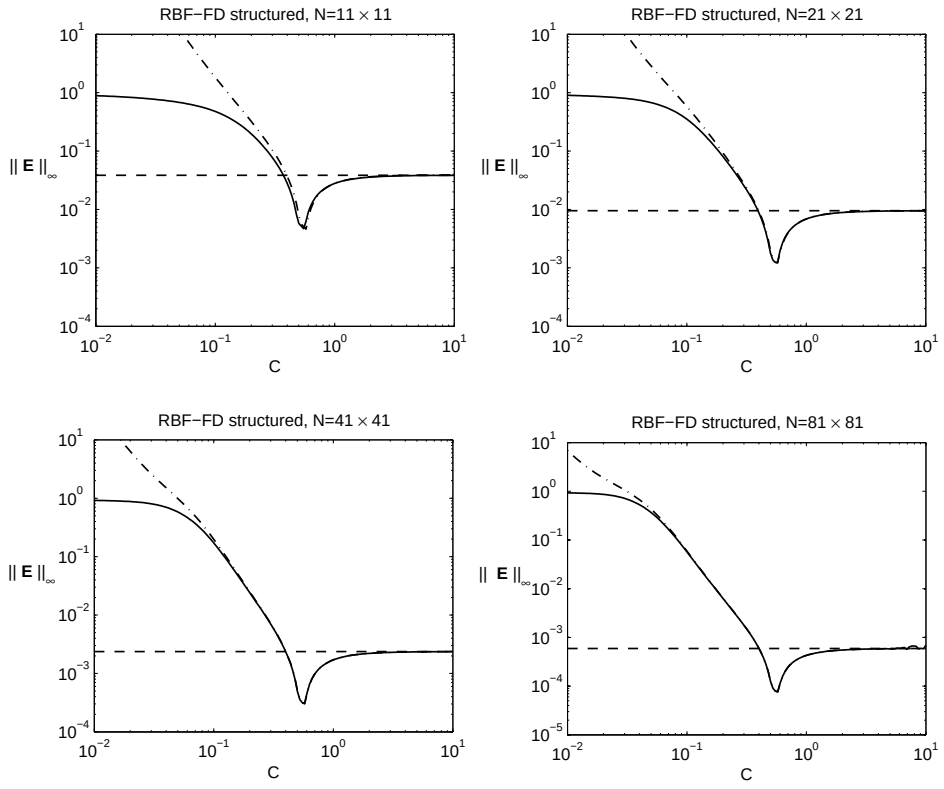


Figure 3.5: Infinite norm of the errors of problem (3.16) with exact solution (3.17) as function of c, using different number N of structured nodes. Solid lines: global RBF-FD error (3.5). Dot-dashed lines: estimated error (2.14). Dashed lines: finite difference error.

this equation, u is approximated using second order central differences, the term $u^{(2,0)} + u^{(0,2)}$ is calculated exactly by substituting for function f, and $u^{(4,0)} + u^{(0,4)} = \Delta f - 2u^{(2,2)}$, where $u^{(2,2)}$ is approximated from $\tilde{u}$ using the corresponding second order central difference scheme.

In Fig. 3.5, we plot with solid lines the infinite norm of the error $||\mathbf{E}(c)||_\infty$ as function of c for different number of nodes N. As was the case for the problems in 1D (Figs. 3.1 and 3.3) the estimated error using (2.14), shown with dot-dashed lines, agrees closely with the actual error when $c \gg h$. The agreement improves as N increases (h decreases). For large values of c the global RBF-FD error coincides with the standard finite difference error (depicted with dashed lines).

Table 3.5 shows the same information as the previous tables but for the

Table 3.5: Optimal shape parameters and the corresponding errors for problem (3.16) with exact solution (3.17) using structured nodes.

N	11×11	21×21	41×41	81×81
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$3.847 \cdot 10^{-2}$	$9.516 \cdot 10^{-3}$	$2.370 \cdot 10^{-3}$	$5.921 \cdot 10^{-4}$
c^*	0.5601	0.5772	0.5807	0.5816
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$4.517 \cdot 10^{-3}$	$1.219 \cdot 10^{-3}$	$3.040 \cdot 10^{-4}$	$7.602 \cdot 10^{-5}$
$\tilde{c}^*$	0.5756	0.5811	0.5816	0.5818
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$6.320 \cdot 10^{-3}$	$1.335 \cdot 10^{-3}$	$3.106 \cdot 10^{-4}$	$7.627 \cdot 10^{-5}$
$\|c^* - \tilde{c}^*\|$	0.0155	0.0039	0.0009	0.0002

solution of problem (3.16). As in previous cases, the estimated optimal shape parameter $\tilde{c}^*$ is very close to the exact optimal shape parameter c^*, and there is a very small loss of accuracy resulting from the use of the estimated value $\tilde{c}^*$ instead of the exact value c^*. Again, the RBF-FD method is more accurate than the standard finite difference method, although in this case, the difference in accuracy is only of one order of magnitude.

This same problem was solved in reference [109]. However, for the local RBF interpolation the authors included a constant term in order to impose the condition that the RBF-FD formulas are exact for constants (see equation (1.25) in Section 1.4). Introducing this additional function in the basis used for local interpolation results in an approximation of the laplacian with a local truncation error,

$$\tau_5(\mathbf{x}, c) = \frac{h^2}{12}\left(u^{(4,0)}(\mathbf{x}) + u^{(0,4)}(\mathbf{x})\right) + \frac{9\,h^2}{8\,c^2}\left(u^{(2,0)}(\mathbf{x}) + u^{(0,2)}(\mathbf{x})\right) + O\left(h^4\, P_2(1/c^2)\right). \tag{3.18}$$

Notice that this expression is slightly different from (2.14). The term proportional to u does not appear. This should be expected since the formula is exact for constant functions. However, this additional degree of freedom has a very small impact on the results as can be appreciated in Fig. 3.6. In this figure we plot the infinite norm of the errors as function of c ($N = 21 \times 21$) when the RBF-FD solutions are computed with and without the constant term in the interpolation basis (dashed and solid lines, respectively). We also plot the estimated errors for these two cases, with and without the constant term in the interpolation basis (dotted and dot-dashed lines, respectively). We observe that the optimal shape parameter is only slightly modified when

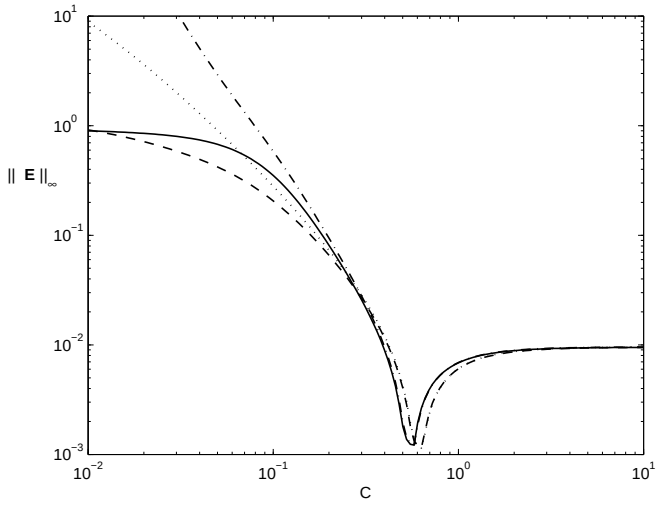


Figure 3.6: Infinite norm of the errors of problem (3.16) with exact solution (3.17), using $N = 21 \times 21$ structured nodes. Solid line: exact error (3.5) using local RBF interpolation with no constant term. Dashed line: exact error (3.5) using local RBF interpolation with constant term. Dot-dashed line: estimated error using (2.14). Dotted line: estimated error using (3.18).

Table 3.6: Same as Table 3.5 but including a constant term in the RBF interpolation, as in [109].

N	6×6	11×11	21×21	51×51
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$1.348 \cdot 10^{-1}$	$3.847 \cdot 10^{-2}$	$9.516 \cdot 10^{-3}$	$1.517 \cdot 10^{-3}$
$c_{[109]}$	1/1.6	1/1.6	1/1.6	1/1.6
$\|\|\mathbf{E}(c_{[109]})\|\|_\infty$	$2.692 \cdot 10^{-2}$	$4.305 \cdot 10^{-3}$	$1.147 \cdot 10^{-3}$	$1.850 \cdot 10^{-4}$
c^*	0.5899	0.6246	0.6143	0.6445
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$1.667 \cdot 10^{-2}$	$4.267 \cdot 10^{-3}$	$1.144 \cdot 10^{-3}$	$1.840 \cdot 10^{-4}$
$\tilde{c}^*$	0.6361	0.6365	0.6176	0.6451
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$2.993 \cdot 10^{-2}$	$5.457 \cdot 10^{-3}$	$1.145 \cdot 10^{-3}$	$1.857 \cdot 10^{-4}$
$\|c^* - \tilde{c}^*\|$	0.0462	0.0119	0.0033	0.0006

the constant term is added to the RBF-FD interpolation.

Table 3.6 shows the same information as Table 3.5, but for the case in which a constant term is added to the interpolation. Row 2 shows the results obtained in reference [109] (see Table 3 of [109]) for the case $c = 1/1.6 = 0.625$, which gives the most accurate results reported in [109] for the 5 node RBF-FD formula. In this table, we have used the same grids used in [109]. As in previous cases, the analytical approximation to the error makes it possible to accurately compute the optimal shape parameter without knowing the exact solution of the problem.

3.4.3.2 Unstructured nodes

Consider now the case in which the domain is discretized using N unstructured nodes. The local truncation error for six unequally spaced nodes $\{(x,y),(x+h,y+\lambda_1 h),(x+\beta_2 h,y+\lambda_2 h),(x+\beta_3 h,y+\lambda_3 h),(x+\beta_4 h,y+\lambda_4 h),(x+\beta_5 h,y+\lambda_5 h)\}$ central difference scheme is

$$\begin{aligned}
\tau_6(\mathbf{x},c) &= h\,\left[A_{0,0}\,u^{(3,0)}(\mathbf{x}) + A_{0,1}\,u^{(2,1)}(\mathbf{x}) \right. \\
&\qquad \left. + A_{0,2}\,u^{(1,2)}(\mathbf{x}) + A_{0,3}\,u^{(0,3)}(\mathbf{x})\right] \\
&+ \frac{h}{c^2}\,\left[A_{1,0},u^{(1,0)}(\mathbf{x}) + A_{1,1}\,u^{(0,1)}(\mathbf{x})\right] \\
&+ h^2\,\left[B_{0,0}\,u^{(4,0)}(\mathbf{x}) + B_{0,1}\,u^{(3,1)}(\mathbf{x}) + B_{0,2}\,u^{(2,2)}(\mathbf{x}) \right. \\
&\qquad \left. + B_{0,3}\,u^{(1,3)}(\mathbf{x}) + B_{0,4}\,u^{(0,4)}(\mathbf{x})\right] \\
&+ \frac{h^2}{c^2}\,\left[B_{1,0},u^{(2,0)}(\mathbf{x}) + B_{1,1}\,u^{(1,1)}(\mathbf{x}) + B_{1,2}\,u^{(0,2)}(\mathbf{x})\right] \\
&+ \frac{h^2}{c^4}\,B_{2,0}\,u(\mathbf{x}) + O\left(h^3\,P_3(1/c^2)\right), \qquad (3.19)
\end{aligned}$$

where the coefficients $A_{i,j}$ and $B_{i,j}$ depend on the surrounding nodes layout $\{\beta_k\}$ and $\{\lambda_k\}$, and its exact values can be computed numerically for each node. In this example, we have not estimated numerically the derivatives of $u(\mathbf{x})$ that appear in equation (3.19). Instead, we have used the exact values of the function and its derivatives in order to analyze the convergence of the error and to estimate the optimal shape parameter.

We will use an unstructured node layout of N^2 nodes: $N^2 - 4(N-1)$ Halton nodes [49] in the interior of the domain and $4(N-1)$ structured nodes on the boundary (see Fig. 3.7). For the local support, we will use stencils with $n = 6$ nodes. For standard finite differences, 6 nodes stencils allow, in principle, a consistent approximation to the laplacian operator

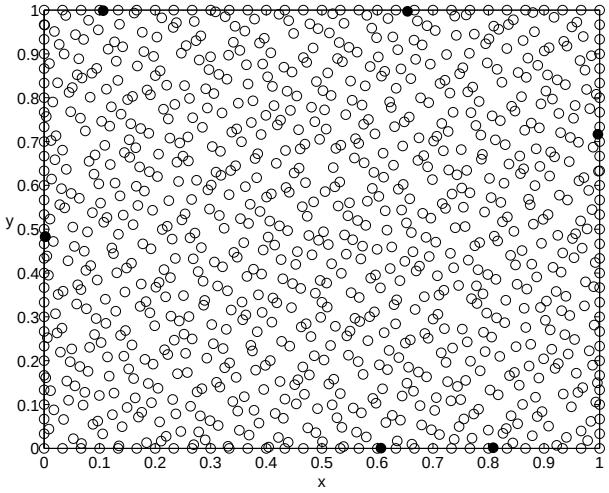


Figure 3.7: Unstructured node layout with 961 nodes: 841 Halton nodes inside the domain and 120 structured nodes on the boundary. Filled circles are the nodes removed from the set after applying Seibold's algorithm

(i.e. the approximation is at least first order accurate) since there are six constraints that have to be satisfied (see Section 2.6). However, there are special configurations of the nodes in the stencil for which there is no solution to the constraints [93], and therefore the coefficients of the finite difference formula can not be computed. The problem of stencil support selection for unstructured nodes is a very crucial topic in finite differences which has been addressed by several authors. In a recent paper, Davydov and Oanh [19] reviewed different support selection methods and proposed a new algorithm based on minimizing the sum of the squares of the angles between two consecutive lines from the central node to the other nodes in the stencil.

We also have found that (at least with Halton nodes) arbitrary stencils using nearest nodes sometimes leads to an ill-conditioned system for large values of c. Therefore, we apply a modified version of the algorithm recently proposed by Seibold [94] to select a valid six node stencil. The algorithm is based on a linear programming approach that guarantees the positivity of the stencil. Since the coefficients of RBF-FD for a given stencil coincide with the coefficients of FD in the limit $c \to \infty$, and since Seibold's algorithm guarantees the positivity of scattered FD stencils, then it also guarantees the positivity of the RBF-FD stencil in that limit. In fact, we have found this to be the case in all the experiments that we have done. Applying Seibold's algorithm to the Halton nodes shown in Fig. 3.7, results in a 6 node stencil

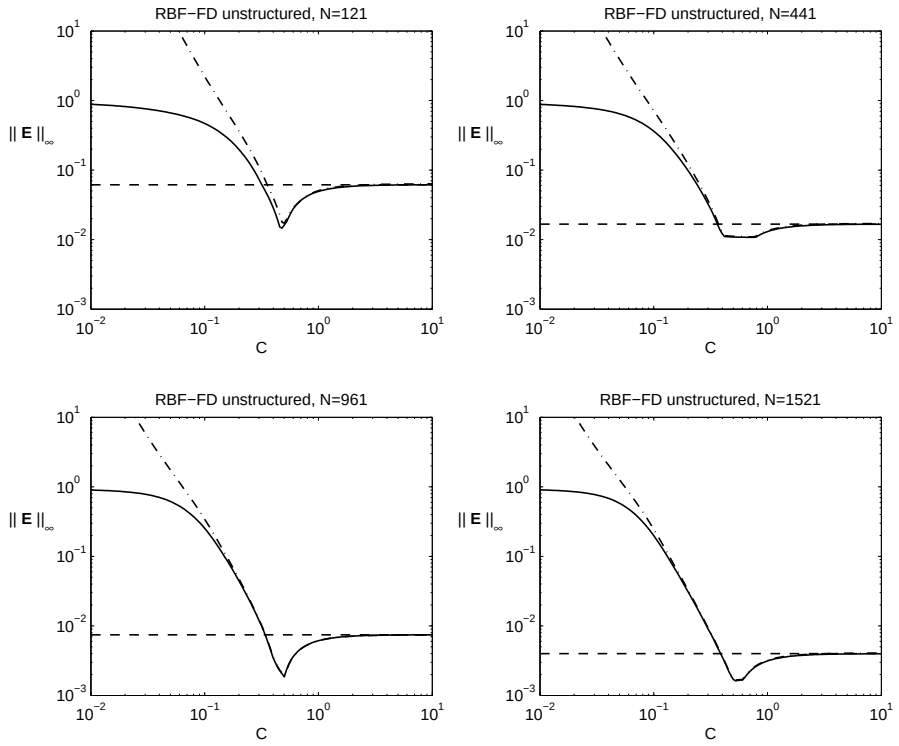


Figure 3.8: Same as Fig. 3.5 but for unstructured nodes.

selection for almost all the interior nodes. There are a few nodes, usually very close to the boundary, for which Seibold's algorithm does not yield a solution. Those nodes are removed from the set (these are shown as filled circles in Fig. 3.7), and Seibold's algorithm is applied again until a valid finite difference 6 node stencil is assigned to each node. Then, steps 2 to 4 of the numerical algorithm described in Section 3.3 are applied in order to compute the optimal RBF-FD solution. Starting from valid finite difference stencils insures the validity of the corresponding RBF-FD stencils.

Plots of the RBF-FD and estimated errors for different number of nodes N appear in Fig. 3.8 with solid and dot-dashed lines, respectively. We observe again an excellent agreement between them. We tabulated the main results in Table 3.7, in which we show the number of nodes in the grid and in parenthesis the number of nodes remaining after applying Seibold's algorithm. The error achieved with the estimated optimal shape parameter $||\mathbf{E}(\tilde{c}^*)||_\infty$ is very close to the optimal one $||\mathbf{E}(c^*)||_\infty$ in all the cases. As expected, the difference between the exact optimal parameter c^* and the

Table 3.7: Same as Table 3.5 but for unstructured nodes.

N	121 (120)	441 (438)	961 (955)	1521 (1513)
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$6.153 \cdot 10^{-2}$	$1.668 \cdot 10^{-2}$	$7.454 \cdot 10^{-3}$	$3.983 \cdot 10^{-3}$
c^*	0.4655	0.7871	0.5027	0.5113
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$1.360 \cdot 10^{-2}$	$1.079 \cdot 10^{-2}$	$1.841 \cdot 10^{-3}$	$1.616 \cdot 10^{-3}$
$\tilde{c}^*$	0.4869	0.7783	0.5058	0.5094
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$1.519 \cdot 10^{-2}$	$1.079 \cdot 10^{-2}$	$1.911 \cdot 10^{-3}$	$1.630 \cdot 10^{-3}$
$\|c^* - \tilde{c}^*\|$	0.0214	0.0088	0.0031	0.0019

estimated one $\tilde{c}^*$ decreases with the number of nodes N.

3.4.3.3 Additional Poisson equation examples

In this section, we address the solution of several problems defined by the Poisson equation which have been proposed in the past. In all cases, we consider equation (3.16) with the function $f(\mathbf{x})$ and $g(\mathbf{x})$ computed, in each case, from the following exact solutions:

$$u_1 = \sin(\pi x)\sin(\pi y) \tag{3.20}$$

$$u_2 = \frac{\arctan\left[2(x+3y-1)\right]}{\arctan\left[2(\sqrt{10}+1)\right]} \tag{3.21}$$

$$u_3 = \frac{3}{4}\, e^{-\frac{(9\,x-2)^2+(9\,y-2)^2}{4}} + \frac{3}{4}\, e^{-\frac{(9\,x+1)^2}{49}-\frac{(9\,y+1)}{10}} \tag{3.22}$$

$$+\frac{1}{2}\, e^{-\frac{(9\,x-7)^2+(9\,y-3)^2}{4}} - \frac{2}{10}\, e^{-(9\,x-4)^2-(9\,x-7)^2} \tag{3.23}$$

$$u_4 = \frac{25}{25+(x-0.2)^2+2y^2}\,. \tag{3.24}$$

Figure 3.9 shows with solid lines the infinite norm of the error $||\mathbf{E}(c)||_\infty$ as a function of c for the four problems considered here (problems (3.20) to (3.24)). In these problems, we have used a regular mesh of 31×31 nodes. Notice that in all cases the estimated errors computed with equation (2.14), shown with dot-dashed lines, are in close agreement with the global RBF-FD errors for $c \gg h$, and that there is always a range of shape parameters for

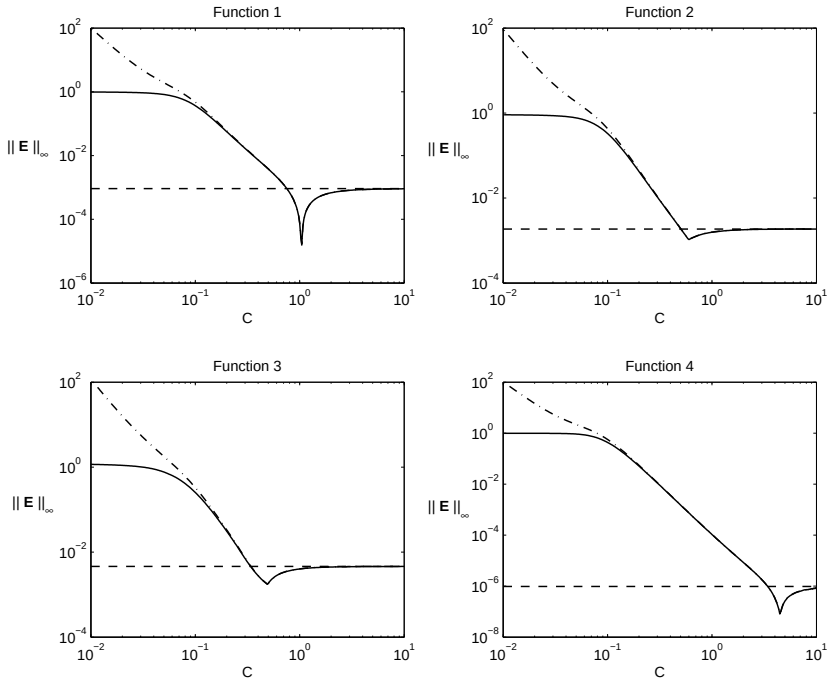


Figure 3.9: Infinite norm of the errors of problem (3.16) with exact solutions (3.20) to (3.24), using $N = 31 \times 31$ structured nodes.

which the RBF-FD solution is more accurate than the standard FD solution (the FD error is shown in dashed lines).

Table 3.8 compares the results obtained with finite differences, and with RBF-FD using the optimal shape parameter computed either from the exact solution or from equation (2.14).

Problem (3.20) was first proposed by Ding *et al.* [20] to numerically analyze the dependence of the approximation error with shape parameter c, inter-nodal distance h, and stencil size n. They used equally spaced nodes in their numerical experiments and concluded that for $n \leq 9$ the error behaves as $||\mathbf{E}|| \approx O((h/c)^{1.9})$. Our analysis in Chapter 2 shows that the error behaves as $||\mathbf{E}|| = O(h^2 P_2(1/c^2))$ (see section 3.2). Problem (3.20) has also been used by Davydov and Oanh [19]. In Fig. 7 of their paper the root mean square (*rms*) error of the numerical differentiation error and the solution error are displayed as a function of c. They use a value of the shape parameter *as large as possible with the RBF matrix still numerically non singular*. Thus, they operate the RBF-FD method in the region where it is equivalent to finite differences, and this is confirmed in Fig. 7 of their paper, where the

Table 3.8: Optimal shape parameters and the corresponding errors for problem (3.16) with exact solutions (3.20) to (3.24), using a 31×31 structured grid.

	u_1	u_2	u_3	u_4
$\|\|\mathbf{E}_{FD}\|\|_\infty$	$9.144 \cdot 10^{-4}$	$1.868 \cdot 10^{-3}$	$4.604 \cdot 10^{-3}$	$9.727 \cdot 10^{-7}$
c^*	1.0380	0.5978	0.4922	4.4949
$\|\|\mathbf{E}(c^*)\|\|_\infty$	$7.092 \cdot 10^{-8}$	$1.043 \cdot 10^{-3}$	$1.742 \cdot 10^{-3}$	$7.436 \cdot 10^{-8}$
$\tilde{c}^*$	1.0387	0.5978	0.4935	4.4957
$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$1.345 \cdot 10^{-6}$	$1.043 \cdot 10^{-3}$	$1.758 \cdot 10^{-3}$	$7.440 \cdot 10^{-8}$
$\|c^* - \tilde{c}^*\|$	0.0007	0.0000	0.0013	0.0008

RBF-FD errors are indistinguishable from those of finite differences. From that figure, and for $N = 31 \times 31$ ($1/N = 10^{-3}$), the *rms* of the numerical differentiation error is approximately $9 \cdot 10^{-3}$ and the rms of the solution error is approximately $4 \cdot 10^{-4}$. Using the optimal shape parameter $\tilde{c}^*$, we obtain an *rms* for the numerical differentiation error of $4.7 \cdot 10^{-7}$, and an *rms* for the solution error of $2.4 \cdot 10^{-8}$. The reason for this extremely high accuracy is that, for this problem, the solution is an eigenfunction of the error and hence, the shape parameter that minimizes the local truncation error (2.14) is independent of location. Thus, the optimal constant shape parameter $\tilde{c}^*$ that we use in our computations minimizes the error at every point of the grid.

Problem (3.21) was used by Larsson and Fornberg [64] to analyze the global RBF method for infinitely smooth radial basis functions ($c \to \infty$). The problem was solved in the unit disk using 50 unstructured nodes. Of the six problems considered in [64], problem (3.21) was the most hard to solve ($||\mathbf{E}||_\infty = 0.23$ for the optimal shape parameter $c^* = 1/0.89 = 1.124$). The results in Fig. 3.9 and Table 3.8 show that, also with the local RBF method, the errors are relatively high. In addition, it can be observed that, with a constant shape parameter, there is very little accuracy increase with respect to finite differences. This can be due to the fact that, either there are many locations for which there is not a local optimal shape parameter, or the optimal local shape parameter varies much with location and there is not a single optimal constant shape parameter c^* which can be successfully used at all locations.

Problem (3.23) was also proposed by Ding *et al.* [20]. The solution is

quite hilly, having three relative extrema and one saddle point within the domain. As in the previous problem, the solution error and the optimal shape parameter are accurately computed with equation (2.14), but there is little improvement with respect to finite differences.

Problem (3.24) was solved in [64] and [109] on the unit disk using unstructured nodes and 9 node stencils. The minimum value of the infinite norm of the solution error with an unstructured set of 200 nodes (shown in Fig. 4 of [109]) is approximately $1.6 \cdot 10^{-5}$. The results in Table 3.8 show that, for a regular rectangular mesh of 961 nodes and 5 node stencils, the infinite norm of the error is $7.440 \cdot 10^{-8}$. The reason for this high accuracy is that function (3.24) is almost constant throughout the domain and, therefore, using a single optimal constant shape parameter for all the nodes results in highly accurate solutions. In fact, in Section 2.5 the optimal shape parameter that minimizes the local approximation error for a nine node stencil at location $(0, 0)$ was $c \approx 1/0.2617 \approx 3.82$ and this result varies very little with location.

3.5 Conclusions

In this chapter we describe how to predict the solution error using the RBF-FD method with a constant value of the shape parameter. It is based on the analytical formulas of the local truncation error derived in Chapter 2 for the MQ RBF-FD method. The idea is that, since the error can be accurately predicted, it is also possible to accurately estimate the optimal shape parameter that minimizes the solution error.

We have described the technique through several examples in 1D (3 node stencils) and 2D (5 and 6 node stencils) using both structured and unstructured nodes. Although it is applied for MQ, it should be pointed out that the same procedure can be followed with any other infinitely smooth RBF (see Table 1.1). For example, the GA RBF-FD weights and its corresponding local truncation error are gathered in Appendix A.

We emphasize that to compute the optimal shape parameter to order $O(h^2)$ it is only necessary to approximate the solution $u(\mathbf{x})$ and certain derivatives to order $O(h^2)$. In practice, this can be achieved by first computing the standard finite difference solution, then use this solution to estimate the optimal shape parameter $\tilde{c}^*$, and finally use this value to compute the optimal RBF-FD solution. From our experience, for two dimensional unstructured grids it is not advisable to estimate derivatives through finite difference formulas, since this will require the selection of appropriate stencils for each cross-derivative in the local truncation error formula. Instead, one can use the RBF global method on a coarse grid and use this solution to

approximate $u(\mathbf{x})$ and the needed derivatives on the unstructured grid.

From the point of view of computational cost the technique proposed requires solving the problem twice; first with standard finite differences and then with the RBF-FD method. Thus, for the solution of a Poisson problem in 2D using N nodes, the method requires the solution of two sparse linear systems of $N \times N$ unknowns. In order to reduce the computational cost, it would be possible to take advantage of the fact that, to leading order, the optimal shape parameter is independent of h. Thus, the finite difference solution or the global RBF solution needed to estimate $\tilde{c}^*$, can be computed in a coarse grid, and then the RBF-FD method can be used to compute the final solution in a fine grid. In this way the computational cost can be significantly reduced.

Chapter 4

Optimal variable shape parameter for multiquadric based RBF-FD method

4.1 Outline

Similarly to what was done on Chapter 3, an algorithm to exploit RBF-FD formulas using a node dependent shape parameter is presented in this chapter. As it is shown, this numerical strategy can give rise to several orders of magnitude increase in accuracy with respect to FD. However, if there are a significant number of nodes for which no optimal value of the shape parameter exists, the improvement in accuracy deteriorates significantly. In these cases, the use of generalized multiquadrics as RBFs may assure the existence of an optimal shape parameter at every node. In this way, we are able to obtain significant accuracy improvements with respect to the optimal constant method for all the example problems analyzed in Chapter 3.

The Chapter is organized as follows. In Section 4.2, we explain how to compute the optimal variable shape parameters. In Section 4.3, we describe the numerical algorithm to compute them. In Section 4.4, we show one dimensional examples using both structured and unstructured grids. In Section 4.5, we use generalized multiquadrics as RBFs to assure the existence of an optimal shape parameter at every node. In Section 4.6, we show one dimensional examples using both structured and unstructured grids. For comparison purposes we use the same examples used in Chapter 3 in all the cases. Finally, Section 4.7 contains our conclusions.

4.2 Optimal Variable Shape Parameter

Consider the Dirichlet problem (3.1) from Section 3.2, which is discretized at an interior node $\mathbf{x}_i$ as

$$\sum_{j=1}^{n} \alpha_{ij}(c_i) u(\mathbf{x}_j) = f(\mathbf{x}_i) + \tau_n(\mathbf{x}_i\,;c_i), \qquad 1 \le i \le N_I, \tag{4.1}$$

where $\tau_n(\mathbf{x}_i\,;c_i)$ is the local truncation error resulting from approximating $\mathcal{L}u$ at $\mathbf{x}_i$ using the RBF-FD method with a stencil of n nodes and c_i is the corresponding node dependent shape parameter. In matrix form, it can be written as

$$A(\mathbf{c})\mathbf{u} = \mathbf{f} + \boldsymbol{\tau}(\mathbf{c}) \ , \tag{4.2}$$

where $A(\mathbf{c})$ is a $N_I \times N_I$ sparse matrix whose entries are the weighting coefficients $\alpha_{ij}(c_i)$, $\mathbf{u}$ is the vector of exact solutions at the interior nodes, $\mathbf{c}$ is the vector of shape parameters at these nodes, and $\boldsymbol{\tau}(\mathbf{c})$ is a vector formed by the corresponding local truncation errors, $\boldsymbol{\tau}(\mathbf{c}) = [\tau_n(\mathbf{x}_1\,;c_1), \ldots, \tau_n(\mathbf{x}_{N_I}\,;c_{N_I})]^T$.

As in Section 3.2, the global RBF-FD error is defined as

$$\mathbf{E}(\mathbf{c}) \equiv \mathbf{u} - \hat{\mathbf{u}}(\mathbf{c}) = A^{-1}(\mathbf{c})\,\boldsymbol{\tau}(\mathbf{c}), \tag{4.3}$$

where $\hat{\mathbf{u}} = A^{-1}(\mathbf{c})\,\mathbf{f}$ is the RBF-FD approximation to the exact solution $\mathbf{u}$. We can state our problem as the problem of finding the vector of shape parameters $\mathbf{c}$ which minimizes (4.3) in a certain norm. In this way, we define the *optimal variable shape parameter* as the vector $\mathbf{c}^*$ such that

$$\|\mathbf{E}(\mathbf{c}^*)\| = \min_{\mathbf{c}} \|\mathbf{E}(\mathbf{c})\| = \min_{\mathbf{c}} \left\|A^{-1}(\mathbf{c})\,\boldsymbol{\tau}(\mathbf{c})\right\|. \tag{4.4}$$

As in Chapter 3, it is not possible to calculate the optimal shape parameter from (4.4) since the local truncation error $\boldsymbol{\tau}(\mathbf{c})$ depends on the exact solution $\mathbf{u}$ and its derivatives. However, these coefficients can be estimated without losing accuracy to leading order using an approximate solution $\tilde{\mathbf{u}}$ instead of the exact solution $\mathbf{u}$. It lets us to obtain an approximate value $\tilde{\mathbf{c}}^*$ to the optimal shape parameter vector $\mathbf{c}^*$ such that

$$||\tilde{\mathbf{E}}(\tilde{\mathbf{c}}^*)|| = \min_{\mathbf{c}} ||A^{-1}(\mathbf{c})\,\tilde{\boldsymbol{\tau}}(\mathbf{c})|| \ , \tag{4.5}$$

where $\tilde{\boldsymbol{\tau}}(\mathbf{c})$ is the estimated local truncation error computed with the numerical approximation $\tilde{\mathbf{u}}$.

Problem (4.5) is the same type of minimization problem (3.9) solved in Chapter 3, but now one has to find N_I unknown shape parameters at the

interior nodes, instead of only one constant shape parameter c as was done in Chapter 3. Furthermore, $||A^{-1}(\mathbf{c})\,\tilde{\boldsymbol{\tau}}(\mathbf{c})||$ is a scalar function of N_I dimensions with an extremely high number of local minima, so minimization algorithms, such as the routine `fminsearch` of MATLAB, are of little use because they immediately fall in one of these local minima.

To compute the global minimum it is necessary to use nonlinear optimization algorithms such as *simulated annealing* [59]. We have used this technique to solve the one dimensional boundary value problem (3.10) considered as a first example in Chapter 3 and we have obtained an error of $||\mathbf{E}(\mathbf{c})||_\infty = 2.947\,10^{-4}$. This result is significantly more accurate than the one obtained with the optimal constant shape parameter ($||\mathbf{E}(c)||_\infty = 1.178\,10^{-3}$). However, this procedure is computationally very expensive and, in general, it is not capable of finding the absolute minimum of the problem.

Thus, instead of solving problem (4.5) to find the vector $\tilde{\mathbf{c}}^*$ that minimizes the estimated global error $||\tilde{\mathbf{E}}(\mathbf{c})||_\infty$, we compute the values $\tilde{\mathbf{c}}^+$ that minimize the estimated local truncation errors $\tilde{\boldsymbol{\tau}}(\mathbf{c})$. This strategy is justified since $||\tilde{\mathbf{E}}(c)|| \leq K\,\|\tilde{\boldsymbol{\tau}}(c)\|$ as $h/c \to 0$, where K is a constant independent of c and h.

Since the *ith*-element of $\tilde{\boldsymbol{\tau}}(\mathbf{c})$, $\tilde{\tau}_n(\mathbf{x}_i\,;\,c_i)$, only depends on the shape parameter c_i, minimizing $|\tilde{\boldsymbol{\tau}}(\mathbf{c})|$ involves N_I minimization problems with one unknown each, which is a much more tractable problem. In fact, we compute the optimal shape parameter $\tilde{c}_i^+$ at each interior node $\mathbf{x}_i$ by solving $\tilde{\tau}_n(\mathbf{x}_i\,;\,\tilde{c}_i^+) = 0$. This is a polynomial equation that can be solved analytically using the explicit formulas for the local errors derived in Chapter 2.

For an optimal shape parameter $\tilde{c}_i^+$ to be valid, two conditions must be satisfied: (i) the solution for $\tilde{\tau}_n(\mathbf{x}_i\,;\,\tilde{c}_i^+) = 0$ must be real, and (ii) $\tilde{c}_i^+ \gg h$. If condition (i) is not satisfied, there is not a value of the shape parameter for which the local approximation error is zero. If condition (ii) is not satisfied, then the optimal value computed is not valid since it is obtained using the local error formulas outside their region of validity.

4.3 Numerical algorithm

The method described in the previous section is implemented as follows:

1. For each interior node $\mathbf{x}_i$ determine a stencil of n surrounding nodes.
2. Use finite differences to compute an approximate solution $\tilde{\mathbf{u}}(\mathbf{x}_i)$.
3. At each interior node $\mathbf{x}_i$, compute the estimated value of the optimal shape parameter, $\tilde{c}_i^+$, using the approximate formulas derived in Chap-

ter 2. These formulas depend on the value of the function and its derivatives at the node, which are estimated using the finite difference solution $\tilde{\mathbf{u}}(\mathbf{x}_i)$.

4. Use (1.18) to compute the RBF-FD coefficients α_{ij} numerically and, therefore, matrix $\mathbf{A}(\tilde{\mathbf{c}}^+)$. In nodes where there is not optimal shape parameter use the standard finite difference coefficients.

5. Compute the optimal RBF-FD approximate solution $\hat{\mathbf{u}}(\tilde{\mathbf{c}}^+) = A^{-1}(\tilde{\mathbf{c}}^+)\,\mathbf{f}$.

4.4 Example problems in one dimension

In this section, we will apply the numerical algorithm just described to the solution of the same example problems in 1D and 2D that were solved in the previous chapter with an optimal constant shape parameter. We will use both structured and non structured nodes and we will show that using the optimal node dependent value of the shape parameter leads to further significant improvements in accuracy.

4.4.1 One dimensional boundary value problem

Consider the problem (3.10) from Section 3.4, which takes the form

$$\begin{cases} u_{xx} = f(x), & 0 < x < 1, \\ u(0) = 1, & u(1) = 1 + \dfrac{\sqrt{2}}{2} \end{cases} \tag{4.6}$$

where $f(x)$ is computed from the known solution $u(x) = 1 - \sin\left(\dfrac{5\pi}{4}x\right)$.

4.4.1.1 Structured nodes

Let us discretize the domain in (4.6) using $N = 41$ structured nodes, and let us use a three node $\{x_i - h, x_i, x_i + h\}$ central difference scheme to approximate the second derivative. The resulting local truncation error (2.11) is approximated as in equation (3.11), yielding

$$\tilde{\tau}_3(x_i; c_i) = \frac{h^2}{12}\, f''(x_i) + \frac{h^2}{c_i^2}\, f(x_i) - \frac{3\,h^2}{4\,c_i^4}\, \tilde{u}(x_i). \tag{4.7}$$

The optimal shape parameters $\tilde{c}_i^+$ are computed equating (4.7) to zero for every node x_i.

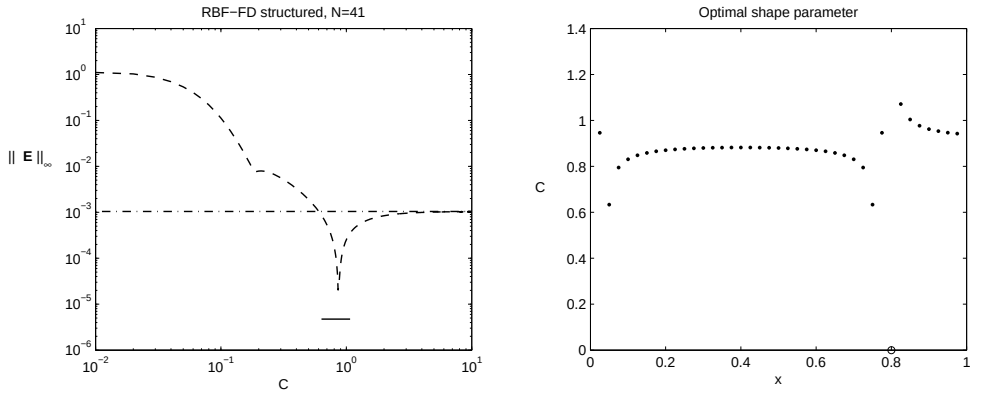


Figure 4.1: Left plot: Infinite norm of the errors of problem (4.6) as function of c, using $N = 41$ structured nodes. Solid line; global RBF-FD error with variable $\tilde{c}_i^+$. Dashed line; global RBF-FD error with constant c. Dot-dashed line; FD error. Right plot: $(\cdot)$ optimal shape parameter distribution $\tilde{\mathbf{c}}^+$. The nodes where $\tilde{c}_i^+$ does not exist are marked with a circle $(\circ)$.

In the left plot of Fig. 4.1 we show with a dashed line the infinity norm of the global RBF-FD error, $||\mathbf{E}(c)||_\infty = ||\mathbf{u} - \hat{\mathbf{u}}(c)||_\infty$, using a constant shape parameter c throughout the domain. Notice that in the limit of increasingly flat basis functions ($c \to \infty$) the RBF-FD error approaches the standard finite difference one [109], shown with a dot-dashed line in the figure. The infinity norm of the RBF-FD error using the algorithm described in Section 4.3 is shown with a solid line. The length of the line represents the range of the optimal values of $\tilde{c}_i^+$. Notice that the accuracy is slightly higher with the variable shape parameter than with the constant one.

The right plot of Fig. 4.1 shows the distribution of the optimal shape parameters. These values are quite close to the value of the optimal constant shape parameter $c \approx 0.8626$. Only near $x = 0$ and $x = 0.8$ the local optimal values $\tilde{c}_i^+$ differ significantly from the constant value. This explains why, in this case, there is not a very significant improvement in accuracy using a variable shape parameter. Also notice that for $x_i = 0.8$ there is not a real value $\tilde{c}_i^+$ that satisfies $\tilde{\tau}_3(x_i; \tilde{c}_i^+) = 0$. These nodes are marked with the symbol $(\circ)$ in the figure.

Table 4.1 shows results for different values of N. The second and third columns display the minimum and maximum values of the optimal shape parameters $\tilde{c}_i^+$, respectively, the forth column the percentage of nodes for which $\tilde{c}_i^+$ does not exist ($\% N$), and the fifth column the infinite norm of the error $||\mathbf{E}(\tilde{\mathbf{c}}^+)||_\infty$. For comparison, we show in the sixth and seventh columns the corresponding results for the optimal constant shape parameter ($\tilde{c}^*$ and

Table 4.1: RBF-FD results for problem (4.6): N is the number of nodes; $\min(\tilde{\mathbf{c}}^+)$ and $\max(\tilde{\mathbf{c}}^+)$ are the minimum and maximum values of the optimal shape parameters $\tilde{c}_i^+$; $\%\,N$ and $||\mathbf{E}(\tilde{\mathbf{c}}^+)||_\infty$ are the percentage of nodes for which c_i^+ does not exist and the corresponding infinite norm of the error; $\tilde{c}^*$ and $||\mathbf{E}(\tilde{c}^*)||_\infty$ are the optimal constant shape parameter and the corresponding infinite norm of the error; $||\mathbf{E}_{FD}||_\infty$ is the infinite norm of the conventional finite differences.

Structured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
21	0.6326	1.0039	4.8	$3.698 \cdot 10^{-5}$	0.8628	$7.763 \cdot 10^{-5}$	$4.181 \cdot 10^{-3}$
41	0.6333	1.0712	2.4	$4.728 \cdot 10^{-6}$	0.8626	$1.735 \cdot 10^{-5}$	$1.044 \cdot 10^{-3}$
61	0.6334	1.1789	1.6	$1.959 \cdot 10^{-6}$	0.8625	$7.541 \cdot 10^{-6}$	$4.638 \cdot 10^{-4}$
81	0.6334	1.3730	1.2	$1.083 \cdot 10^{-6}$	0.8625	$4.216 \cdot 10^{-6}$	$2.609 \cdot 10^{-4}$
Unstructured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
21	0.2760	1.0596	0	$2.211 \cdot 10^{-4}$	0.7858	$1.207 \cdot 10^{-3}$	$4.928 \cdot 10^{-3}$
41	0.3962	1.3382	0	$2.625 \cdot 10^{-6}$	0.7959	$1.981 \cdot 10^{-4}$	$1.327 \cdot 10^{-3}$
61	0.3965	1.4689	0	$2.516 \cdot 10^{-6}$	1.0069	$4.586 \cdot 10^{-5}$	$3.818 \cdot 10^{-4}$
81	0.3715	1.7317	1.2	$3.348 \cdot 10^{-6}$	0.8071	$2.595 \cdot 10^{-5}$	$3.377 \cdot 10^{-4}$

$||\mathbf{E}(\tilde{c}^*)||_\infty$, respectively), and in the last column the results for conventional finite differences ($\|\mathbf{E}_{FD}\|_\infty$). For all values of N, there is some improvement in accuracy using the variable optimal shape parameter instead of the constant optimal one.

4.4.1.2 Unstructured nodes

When the domain is discretized with unequally spaced nodes, the local truncation error using a three node $\{x_i - h, x_i, x_i + \lambda h\}$ central difference scheme is only of order $O(h)$ (see Section 2.6 in Chapter 2). To improve the accuracy, terms of order $O(h^2)$ are included in the formula as in equation (3.12).

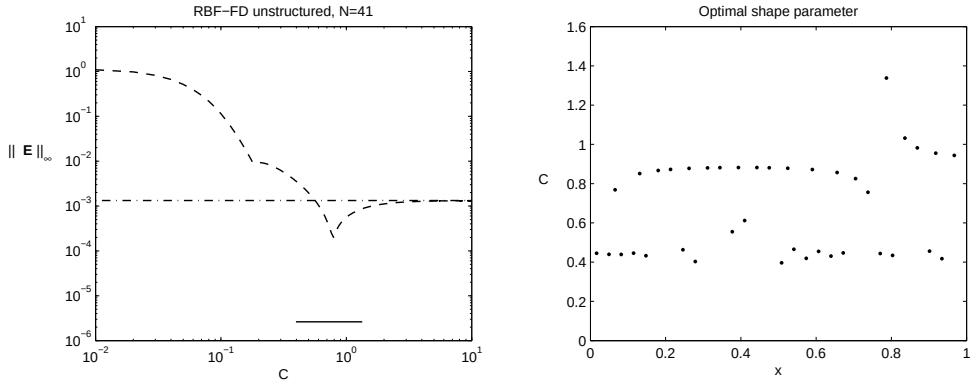


Figure 4.2: Same as Fig. 4.1 but for unstructured nodes.

Therefore, the estimated local error is

$$\begin{aligned}\tilde{\tau}_3(x_i;c_i) &= \frac{\lambda-1}{3}\,h\,f'(x_i) + (\lambda-1)\,\frac{h}{c_i^2}\,\tilde{u}'(x_i) \\ &+ [\lambda\,(\lambda-1)+1]\,\frac{h^2}{12}\,f''(x_i) + \lambda\frac{h^2}{c_i^2}\,f(x_i) \\ &+ [\lambda\,(\lambda-5)+1]\,\frac{h^2}{4\,c_i^4}\,\tilde{u}(x_i)\,,\end{aligned} \tag{4.8}$$

where we have used (4.6) to replace $u^{(k)}$, $k \geq 2$, by the function f and its derivatives. We have also replaced the exact solution u and its first derivative u' by the first order finite difference approximations $\tilde{u}$ and $\tilde{u}'$, respectively. In this case, $\tilde{u}'$ is computed from $\tilde{u}$ using a first order finite difference approximation. This procedure does not increase the error to leading order.

In Fig. 4.2, we show the results from problem (4.6) using $N = 41$ unstructured nodes. The left plot shows the infinity norms of the global RBF-FD error using a constant shape parameter (dashed line), and using the algorithm described in Section 4.3 (solid line). The length of the solid line represents the range of the optimal variable values $\tilde{c}_i^+$. In this case, there is an improvement of approximately two orders of magnitude between the results for a constant shape parameter and the results for a variable one. The right plot of the figure shows the distribution of optimal shape parameters, which range from 0.4 to 1.4. The optimal constant shape parameter is, in this case, $c^* \approx 0.7959$ (see Table 3.2 in Section 3.4) which is a kind of average of the optimal variable values shown in the figure.

The results for different values of N are summarized in the bottom part of Table 4.1. For all the resolutions reported here, there is an order of magnitude

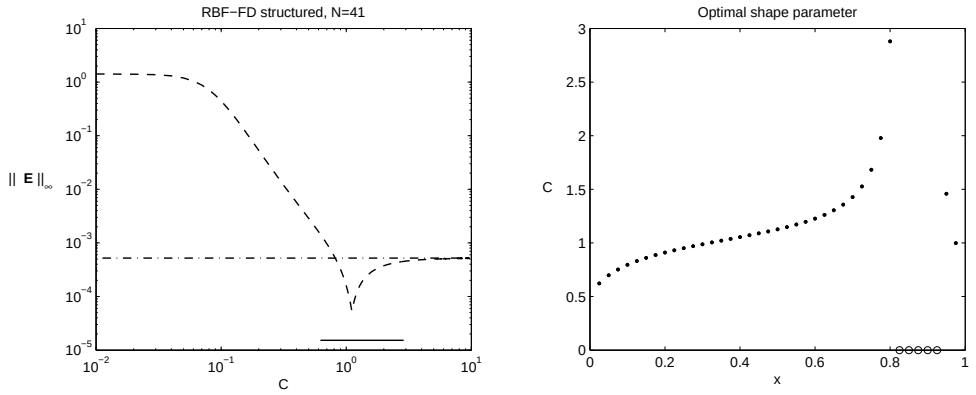


Figure 4.3: Same as Fig. 4.1 but for problem (4.9).

of improvement with respect to the results obtained with a constant optimal shape parameter. This improvement is due to the fact that an optimal value $\tilde{c}_i^+$ exists for all nodes (except for $N = 81$).

4.4.2 Steady convection-diffusion problem

Consider the problem (3.13) from Section 3.4, given by

$$\begin{cases} u_x - u_{xx} = \pi^2 \sin(\pi x) + \pi \cos(\pi x), & 0 < x < 1, \\ u(0) = 0, \qquad u(1) = 1 \end{cases} \tag{4.9}$$

whose exact solution is $u(x) = \sin(\pi x) + \dfrac{e^x - 1}{e - 1}$. This problem was proposed and solved in [11].

4.4.2.1 Structured nodes

The local truncation error for the convection-diffusion differential operator (4.9) is given by equation (3.14), which it is estimated as explained in Section 3.4.2.

In Fig. 4.3, we show the results from problem (4.9) using 41 nodes. The left plot shows the infinity norm of the global RBF-FD error using a constant shape parameter (dashed line) and the infinity norm of the global RBF-FD error using the algorithm described in Section 4.3 (solid line). The length of this line represents the range of the optimal values of the shape parameters $\tilde{c}_i^+$ (between 0.6 and 3, approximately). The results for $N = 41$ and for other values of N are summarized in Table 4.2. Notice that there is only a small

Table 4.2: Same as Table 4.1 but for the steady convection-diffusion problem (4.9).

Structured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
11	0.7963	2.8808	9.0	$1.719 \cdot 10^{-4}$	1.1139	$9.245 \cdot 10^{-4}$	$8.337 \cdot 10^{-3}$
21	0.6979	2.8796	9.5	$5.643 \cdot 10^{-5}$	1.1127	$2.161 \cdot 10^{-4}$	$2.088 \cdot 10^{-3}$
41	0.6224	2.8793	12.2	$1.523 \cdot 10^{-5}$	1.1123	$5.282 \cdot 10^{-5}$	$5.220 \cdot 10^{-4}$
81	0.5707	5.8510	11.1	$3.834 \cdot 10^{-6}$	1.1123	$1.314 \cdot 10^{-5}$	$1.305 \cdot 10^{-4}$
Unstructured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
11	0.9933	3.3864	54.5	$1.008 \cdot 10^{-2}$	1.1409	$5.060 \cdot 10^{-3}$	$1.105 \cdot 10^{-2}$
21	0.8469	3.2638	57.1	$1.525 \cdot 10^{-3}$	0.9818	$4.832 \cdot 10^{-4}$	$2.667 \cdot 10^{-3}$
41	0.6074	2.2497	31.7	$2.843 \cdot 10^{-4}$	1.0536	$5.979 \cdot 10^{-5}$	$6.788 \cdot 10^{-4}$
81	0.4002	3.9330	25.9	$1.026 \cdot 10^{-4}$	1.0539	$1.584 \cdot 10^{-5}$	$1.685 \cdot 10^{-4}$

improvement with respect to the results obtained with the optimal constant value of the shape parameter ($c^* \approx 1.1123$). The reason for these results can be explained by looking at the optimal shape parameter distribution shown in the right plot of this figure. Observe that there are five nodes, in the vicinity of $x = 0.85$, for which no optimal shape parameter exists. In these nodes, standard finite difference formulas are used to approximate the convection-diffusion differential operator. This approximation deteriorates significantly the overall accuracy. Similar results are obtained for other values of N, as can be seen in Table 4.2. In fact, we have observed that the accuracy obtained with the algorithm described in Section 4.3 is highly dependent on the number of nodes for which $\tilde{c}_i^+$ exists. If there are very few nodes for which $\tilde{c}_i^+$ does not exist, then the improvement of the accuracy compared to standard FD is very high. However, if there are many nodes for which $\tilde{c}_i^+$ does not exist then the accuracy is similar to that obtained with an optimal constant c^* or even with standard FD formulas.

In this way, consider for instance the steady convection-diffusion problem

$$\begin{cases} u_x - u_{xx} = 0, & 0 < x < 1, \\ u(0) = 0, \qquad u(1) = 1\,, \end{cases} \tag{4.10}$$

whose exact solution is $u(x) = \dfrac{e^x - 1}{e - 1}$. This problem was also proposed and

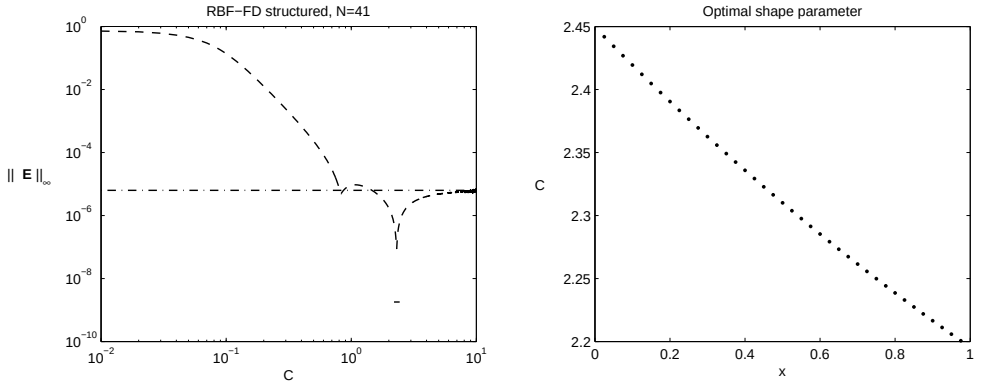


Figure 4.4: Same as Fig. 4.1 but for problem (4.10).

solved in [11].

The left plot of Fig. 4.4 shows the corresponding errors for this problem. Notice that, in this case, there is a very significant improvement of two orders of magnitude in accuracy. The reason for this improvement can be understood by considering the optimal shape parameter distribution which is shown in the right plot of Fig. 4.4. For all the nodes there is a value $\tilde{c}_i^+$ of the shape parameter which cancels out the local truncation error (3.14). It is remarkable that although the values of $\tilde{c}_i^+$ are relatively constant ($2.20 \leq \tilde{c}_i^+ \leq 2.44$, for all i), those small variations suffice to produce a considerable increase in accuracy. Similar accuracy improvements are obtained for other values of N (see Table 4.3) since in all cases there exits an optimal value $\tilde{c}_i^+$ in each node ($\% N = 0$).

4.4.2.2 Unstructured nodes

In the case that the domain is discretized with unequally spaced nodes, the estimated local truncation error using a three node $\{x-h, x, x+\lambda h\}$ central difference scheme for the convection-diffusion operator is given by equation (3.15). Function $\tilde{u}$ and its derivatives are approximated as described in Section 3.4.2.

In Fig. 4.5, we show the results from problem (4.9) using 41 unstructured Halton nodes. The left plot shows the infinity norm of the global RBF-FD error using a constant shape parameter (dashed line) and the infinity norm of the global RBF-FD error using the algorithm described in Section 4.3 (solid line). The length of the line shows the range of the optimal values of the shape parameters $\tilde{c}_i^+$ (between approximately 0.6 and 2.3). The right plot shows the optimal shape parameter distribution. Notice that there is a significant

Table 4.3: Same as Table 4.1 but for the steady convection-diffusion problem (4.10).

Structured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\, N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
11	2.2171	2.4195	0	$3.466 \cdot 10^{-7}$	2.3065	$1.463 \cdot 10^{-6}$	$1.007 \cdot 10^{-4}$
21	2.2060	2.4343	0	$2.175 \cdot 10^{-8}$	2.3066	$3.266 \cdot 10^{-7}$	$2.515 \cdot 10^{-5}$
41	2.2006	2.4419	0	$1.793 \cdot 10^{-9}$	2.3065	$7.786 \cdot 10^{-8}$	$6.292 \cdot 10^{-6}$
61	2.1988	2.4444	0	$1.230 \cdot 10^{-9}$	2.3065	$3.418 \cdot 10^{-8}$	$2.797 \cdot 10^{-6}$
Unstructured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\, N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}_{FD}\|\|_\infty$
11	2.2125	2.3608	54.5	$3.812 \cdot 10^{-4}$	0.9336	$3.8526 \cdot 10^{-4}$	$3.907 \cdot 10^{-4}$
21	2.2101	2.4084	52.3	$3.397 \cdot 10^{-5}$	3.3788	$4.093 \cdot 10^{-5}$	$4.722 \cdot 10^{-5}$
41	2.2023	2.4297	46.3	$5.099 \cdot 10^{-6}$	2.4715	$5.216 \cdot 10^{-6}$	$9.431 \cdot 10^{-6}$
61	2.1988	2.4445	3.3	$3.325 \cdot 10^{-6}$	1.7107	$4.107 \cdot 10^{-6}$	$5.723 \cdot 10^{-6}$

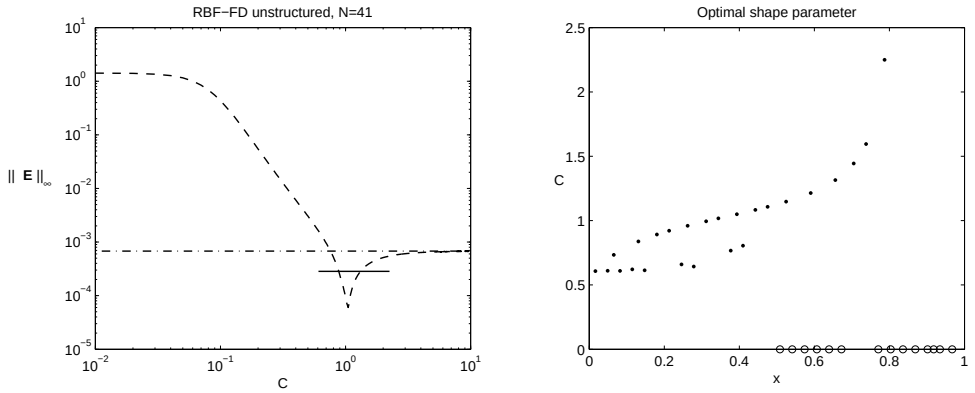


Figure 4.5: Same as Fig. 4.3 but for 41 unstructured Halton nodes.

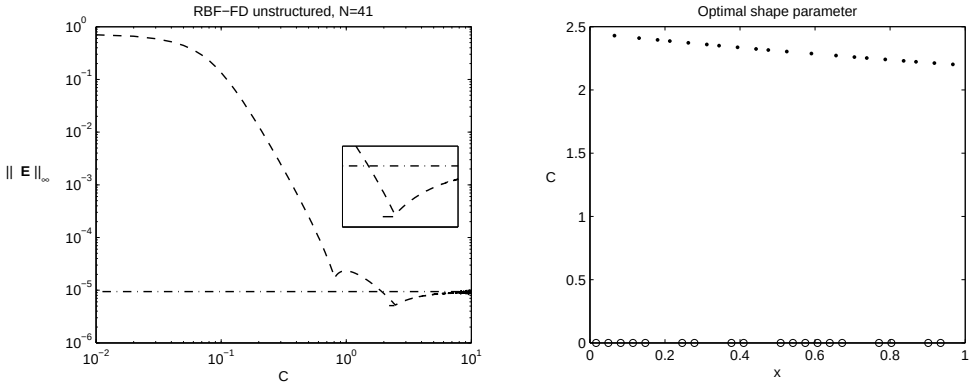


Figure 4.6: Same as Fig. 4.4 but for 41 unstructured Halton nodes.

number of nodes for which an optimal value of the shape parameter $\tilde{c}_i^+$ does not exist (31.7 %) and, therefore, the standard finite difference approximation is used at those nodes. As a result, the overall accuracy of the solution is degraded in comparison with the optimal constant shape parameter. Similar results are obtained for other values of N, as can be seen in the bottom part of Table 4.2.

Figure 4.6 shows the corresponding results for problem (4.10). Observe in the right plot of this figure that for approximately half of the nodes (46.3 %) an optimal value $\tilde{c}_i^+$ exists ($2.2 \leq \tilde{c}_i^+ \leq 2.45$), while for the other half no $\tilde{c}_i^+$ exists. The resulting overall accuracy is very similar to that obtained with the optimal constant shape parameter technique. Similar results are obtained for other values of N as can be seen in the bottom part of Table 4.3.

It should be mentioned that in some of the nodes marked with the symbol ($\circ$), there are values of the shape parameter for which the local truncation errors (3.15) are minimum. However, the resulting shape parameters are small, and do not satisfy the assumption $c \gg h$ for which the formulas are valid. Thus, for those values of c, Eq. (3.15) is not a good approximation to the local truncation error and, therefore, it can not be used to compute a valid $\tilde{c}_i^+$. Accordingly, we only accept the $\tilde{c}_i^+$ values that satisfy the condition $\tilde{c}_i^+ > c_{min} \gg h$, where c_{min} is a previously defined threshold. Hence, item 4 of the numerical algorithm in Section 3 is substituted by

4. Use (1.18) to compute numerically the RBF-FD coefficients α_{ij} and therefore matrix $\mathbf{A}(\tilde{\mathbf{c}}^+)$. In nodes where there is not an optimal shape parameter or where $\tilde{c}_i^+ < c_{min}$, use standard finite difference coefficients.

In Figure 4.6 and Table 4.3 we have used $c_{min} = 0.5$. We will use this modified algorithm in the rest of the chapter.

4.5 Generalized multiquadrics.

From the previous results, it is apparent that the use of a variable shape parameter can give rise to several orders more accurate solutions if a valid $\tilde{c}_i^+$ exists at almost all nodes of the computational domain. However, very often, there are nodes for which no $\tilde{c}_i^+ \gg h$ exists for which the leading order of the RBF-FD multiquadric based approximation (Eqs. (3.14) and (3.15)) is zero. Hence, we propose the use of another RBF that ensures that the local truncation error is zero to leading order at all nodes. To this end, one possibility that we have successfully used is the generalized multiquadric

$$\phi(\|\mathbf{x}-\mathbf{x}_k\|, c, \beta) = \left(c^2 + \|\mathbf{x}-\mathbf{x}_k\|\right)^{\beta/2}, \tag{4.11}$$

where the new (node-dependent) parameter β is chosen so that an optimal value $c_i^+ \gg h$ exists at every node. For the steady convection-diffusion operator, the local truncation error formula analogous to (3.15) is

$$\begin{aligned}
\tau_3(x_i; c_i, \beta_i) &= \frac{(1-\lambda)}{3} h\, u'''(x_i) + \frac{h}{c_i^2}(\beta_i - 2)(\lambda - 1)\, u'(x_i) \\
&+ \frac{h^2}{12}\left[2\lambda\, u'''(x_i) - [\lambda(\lambda-1)+1]\, u^{(IV)}(x_i)\right] \\
&+ \frac{h^2}{c_i^2}\left[\frac{(2-\beta_i)}{2}\lambda u'(x_i) + \right. \\
&\qquad \left. + \frac{(\beta_i - 2)}{6(\beta_i - 3)}\left(\beta_i\left(\lambda^2+\lambda+1\right) - \lambda(\lambda+13) - 1\right) u''(x_i)\right] \\
&+ \frac{h^2}{c_i^4}\frac{(\beta_i-7)(\beta_i-2)}{12(\beta_i-3)}\beta_i\,[\lambda(\lambda-5)+1]\, u(x_i) + O\left(h^3 P_2(1/c_i^2)\right).
\end{aligned} \tag{4.12}$$

The error is now a function of h, λ, c_i and β_i. The objective is to find at each node x_i a valid combination of values (c_i^+, β_i^+) for which the leading order of the local truncation error is zero. For a given β_i equating to zero the leading order equation for the local error (4.13) results in a polynomial equation of second degree in the variable $1/c_i^2$ that we solve analytically. This guarantees, in general, two branches of solutions $c_i^+ = c_i^+(\beta_i^+)$. Hence, there are an infinite number of possible combinations (c_i^+, β_i^+) that make the leading order of the local truncation error zero. However, only those for which

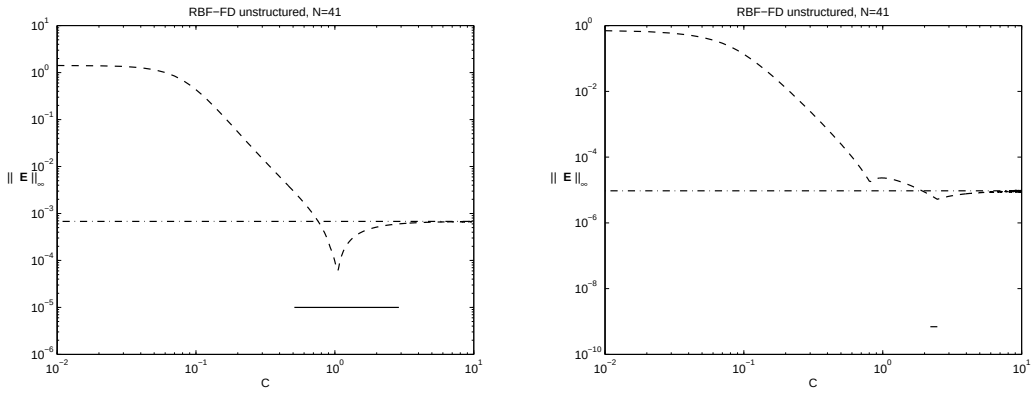


Figure 4.7: Results for the convection-diffusion problems using the generalized multiquadric (4.11). Left: same as Fig. 4.5 (problem (4.9)). Right: same as Fig. 4.6 (problem (4.10)).

c_i^+ is real and $c_i^+ \gg h$ are valid. Between all these possible combinations, we only accept the minimum integer value $\left|\beta_i^+\right|$ which satisfies c_i^+ real and $c_i^+ > c_{min}$. Other strategies, as for instance allowing β_i^+ to be a continuous real variable, are also possible and give the same degree of accuracy.

The left plot of Fig. 4.7 shows the error of the solution of problem (4.9) with 41 unstructured Halton nodes, using (4.11) instead of a multiquadric RBF and $c_{min} = 0.5$. The left plot of Fig. 4.8 shows the corresponding optimal shape parameter distribution $\left(\tilde{c}_\beta^+\right)_i$ and the right side of Fig. 4.8 the optimal β_i^+ distribution. These results should be compared to those obtained with the standard multiquadric RBF which are shown in the right plot of Fig. 4.5. Using the generalized multiquadric (4.11), a valid value of $\left(\tilde{c}_\beta^+\right)_i$ exists for all nodes and this leads to a significant increase in accuracy with respect to the optimal constant shape parameter result. In fact, using (4.11) the resulting error is $||\mathbf{E}(\,\tilde{\mathbf{c}}_\beta^+\,)||_\infty = 1.001 \cdot 10^{-5}$, while using the standard multiquadric (1.3) the error is $||\mathbf{E}(\,\tilde{\mathbf{c}}^+\,)||_\infty = 2.843 \cdot 10^{-4}$. Table 4.4 shows the corresponding results for other values of N.

The right plot of Fig. 4.7 shows the error in the solution of problem (4.10) with 41 unstructured Halton nodes, using (4.11) instead of a multiquadric RBF and $c_{min} = 0.5$. Again, a valid value of c_i^+ exists for all nodes and this leads to an even larger improvement in accuracy with respect to the results obtained with standard multiquadrics: $||\mathbf{E}(\,\tilde{\mathbf{c}}_\beta^+\,)||_\infty = 6.930 \cdot 10^{-10}$ instead of $||\mathbf{E}(\,\tilde{\mathbf{c}}^+\,)||_\infty = 5.099 \cdot 10^{-6}$. This represents four orders of magnitude increase in accuracy.

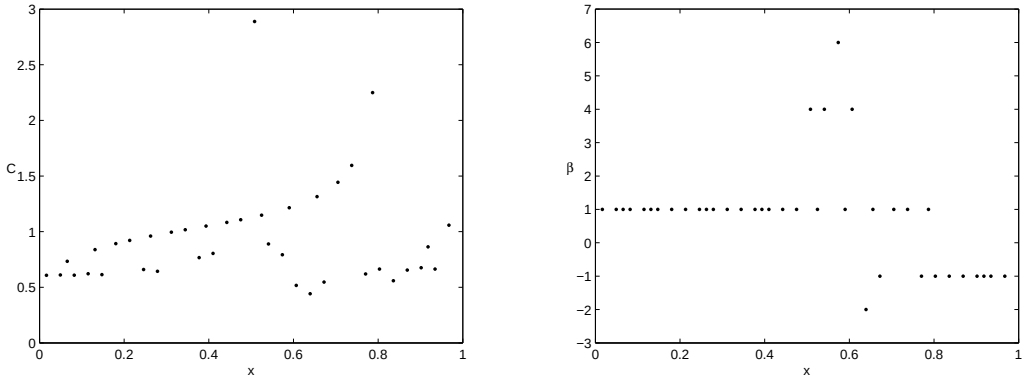


Figure 4.8: Optimal shape parameter distribution $\tilde{c}_\beta^+$ (left) and β^+ parameter distribution (right) for the convection-diffusion problem (4.9) using the generalized multiquadric (4.11).

Table 4.4: RBF-FD results for the steady convection-diffusion problem (4.9) with the generalized multiquadric: N is the number of nodes; $\%\, N_\beta$ and $||\mathbf{E}(\,\tilde{\mathbf{c}}_\beta^+\,)||_\infty$ are the percentage of nodes for which $(c_\beta^+)_i$ does not exist and the corresponding infinite norm of the error; $\%\, N_\beta$ and $||\mathbf{E}(\tilde{c}^+)||_\infty$ are the percentage of nodes for which c_i^+ does not exist and the corresponding infinite norm of the error; $||\mathbf{E}(\tilde{c}^*)||_\infty$ is the infinite norm of the error using an optimal constant shape parameter $\tilde{c}^*$.

Unstructured nodes (beta variable)					
N	$\%\, N_\beta$	$\|\|\mathbf{E}(\,\tilde{\mathbf{c}}_\beta^+\,)\|\|_\infty$	$\%\, N$	$\|\|\mathbf{E}(\,\tilde{\mathbf{c}}^+\,)\|\|_\infty$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$
11	0	$8.860 \cdot 10^{-4}$	54.5	$1.008 \cdot 10^{-2}$	$5.060 \cdot 10^{-3}$
21	0	$1.443 \cdot 10^{-4}$	57.1	$1.525 \cdot 10^{-3}$	$4.832 \cdot 10^{-4}$
41	0	$1.001 \cdot 10^{-5}$	31.7	$2.843 \cdot 10^{-4}$	$5.979 \cdot 10^{-5}$
81	0	$4.623 \cdot 10^{-7}$	25.9	$1.026 \cdot 10^{-4}$	$1.584 \cdot 10^{-5}$

4.6 Example problems in two dimensions

Consider now the two dimensional Poisson problem (3.16) from Section 3.4, which takes the form

$$\begin{cases} \Delta u(\mathbf{x}) &= f(\mathbf{x}), & \text{in } \Omega = (0,1)\times(0,1) \\ u(\mathbf{x}) &= g(\mathbf{x}), & \text{on } \partial\Omega \end{cases} \tag{4.13}$$

where $f(\mathbf{x})$ and $g(\mathbf{x})$ are obtained from the exact solution

$$u = \exp\left[-\left(x-\frac{1}{4}\right)^2 - \left(y-\frac{1}{2}\right)^2\right]\cos(2\,\pi\,y)\,\sin(\pi\,x)\ . \tag{4.14}$$

This problem has been used by Wright and Fornberg [109] to test the performance of the local RBF-FD and local RBF-HFD (Hermite RBF) methods. We have also used it as an example problem to test the performance of the RBF-FD method using an optimal constant shape parameter

4.6.1 Structured nodes

Suppose the domain is discretized using an $N \times N$ structured nodes. Using a five node $\{(x,y),(x-h,y),(x+h,y),(x,y-h),(x,y+h)\}$ scheme, the local truncation error is given by equation (2.14), where u and its derivatives are approximated as it is explained in Section 3.4.

In Fig. 4.9, we show the results from problem (4.13)-(4.14) using 41×41 structured nodes and the standard multiquadrics. The left plot shows, with a dashed line, the infinity norm of the global RBF-FD error using a constant shape parameter throughout the domain. The infinity norm of the global RBF-FD error using the algorithm described in Section 4.3 is shown with a solid line. The length of this line represents the range of the values of the shape parameters $\tilde{c}_i^+$. Although for most nodes, the values of the $\tilde{c}_i^+$ are close to the optimal constant shape parameter $c^* \approx 0.5818$ (see Table 4.5), the accuracy with the variable shape parameter is significantly higher than with the optimal constant one.

The right plot of the figure shows the distribution of the optimal shape parameters $\tilde{c}_i^+$. In this figure, there are two small intervals in the vicinities of $x = 0.25$ and $x = 0.75$ for which $\tilde{c}_i^+$ does not exist. In Fig. 4.10, we represent the corresponding absolute global RBF-FD error distribution for this example. Notice that the maximum error is located at these nodes.

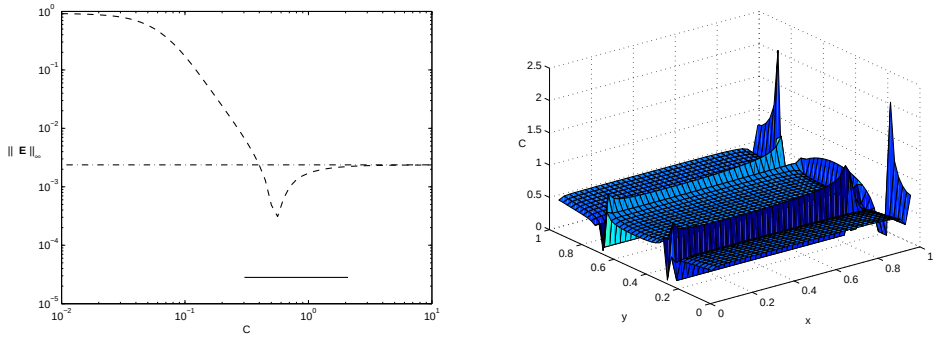


Figure 4.9: Left: infinite norm of the errors of problem (4.13) with exact solution (4.14) as function of c, using $N = 41 \times 41$ structured nodes. Solid line; global RBF-FD error with optimal variable $\mathbf{c}$. Dashed line; global RBF-FD error with constant c. Dot-dashed line; FD error. Right: optimal shape parameter distribution $\tilde{\mathbf{c}}^+$.

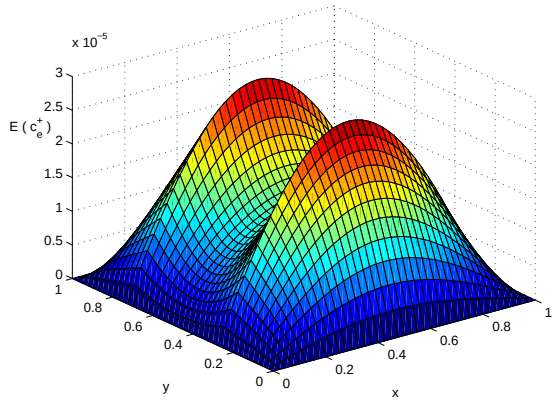


Figure 4.10: Absolute global RBF-FD error of problem (4.13) computed with the optimal shape parameter distribution $\tilde{\mathbf{c}}^+$.

Table 4.5: Same as Table 4.1 but for the laplacian problem (4.13)-(4.14).

Structured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}\|\|_\infty$
11×11	0.4127	0.7979	1.7	$2.422 \cdot 10^{-3}$	0.5779	$6.581 \cdot 10^{-3}$	$3.847 \cdot 10^{-2}$
21×21	0.3317	0.8835	0.5	$1.498 \cdot 10^{-4}$	0.5816	$1.352 \cdot 10^{-3}$	$9.516 \cdot 10^{-3}$
31×31	0.2146	6.1387	0.2	$1.171 \cdot 10^{-4}$	0.5816	$5.641 \cdot 10^{-4}$	$4.220 \cdot 10^{-3}$
41×41	0.1680	2.0905	4.3	$2.810 \cdot 10^{-5}$	0.5818	$3.121 \cdot 10^{-4}$	$2.370 \cdot 10^{-3}$
Unstructured nodes							
N	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%\,N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}\|\|_\infty$
121 (120)	0.1220	3.1054	41.3	$1.415 \cdot 10^{-2}$	0.4869	$1.519 \cdot 10^{-2}$	$6.153 \cdot 10^{-2}$
441 (438)	0.1141	3.2097	30.6	$7.616 \cdot 10^{-3}$	0.7783	$1.079 \cdot 10^{-2}$	$1.668 \cdot 10^{-2}$
961 (955)	0.1041	9.0912	26.5	$5.796 \cdot 10^{-3}$	0.5058	$1.911 \cdot 10^{-3}$	$7.454 \cdot 10^{-3}$
1521 (1513)	0.1001	2.9971	29.3	$5.239 \cdot 10^{-3}$	0.5094	$1.630 \cdot 10^{-3}$	$3.983 \cdot 10^{-3}$

4.6.2 Unstructured nodes

Consider now the case in which the domain is discretized using N unstructured nodes. The local truncation error for six unequally spaced nodes $\{(x,y),(x+h,y+\lambda_1 h),(x+\gamma_2 h,y+\lambda_2 h),(x+\gamma_3 h,y+\lambda_3 h),(x+\gamma_4 h,y+\lambda_4 h),(x+\gamma_5 h,y+\lambda_5 h)\}$ central difference scheme is given by equation (3.19), which is estimated as it is explained in Section 3.4. Figure 4.11 shows the infinite norms of the RBF-FD errors of problem (4.13)-(4.14) using a constant shape parameter (dashed line), and using the algorithm described is Section 4.3 (bar). The gray scale in the bar is proportional to the number of nodes with that optimal shape parameter $\tilde{c}_i^+$. In this case, we have used 21×21 unstructured Halton nodes. Notice that there is a slight improvement of accuracy with respect to the optimal constant RBF-FD solution. For more points the accuracy does not improve (see Table 4.5).

4.6.3 Additional Poisson equation examples

In this section, we address the solution of the additional Poisson equation examples proposed in Section 3.4, where we consider Eq. (4.13) with the

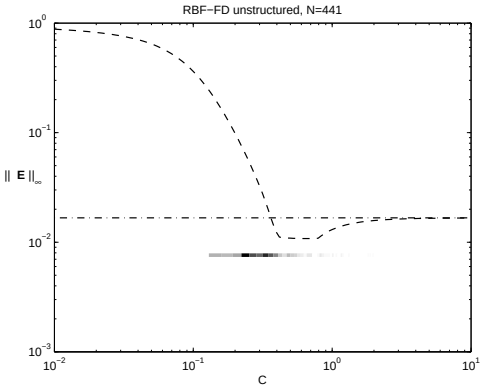


Figure 4.11: Same as Fig. 4.9 but for $N = 441$ unstructured nodes.

function f computed, in each case, from the following exact solutions:

$$u_1 = \sin(\pi x)\sin(\pi y)\,, \tag{4.15}$$

$$u_2 = \frac{\arctan\left[2(x+3y-1)\right]}{\arctan\left[2(\sqrt{10}+1)\right]}\,, \tag{4.16}$$

$$u_3 = \frac{3}{4}\,e^{-\frac{(9\,x-2)^2+(9\,y-2)^2}{4}} + \frac{3}{4}\,e^{-\frac{(9\,x+1)^2}{49}-\frac{(9\,y+1)}{10}} + \\ + \frac{1}{2}\,e^{-\frac{(9\,x-7)^2+(9\,y-3)^2}{4}} - \frac{2}{10}\,e^{-(9\,x-4)^2-(9\,x-7)^2}, \tag{4.17}$$

$$u_4 = \frac{25}{25+(x-0.2)^2+2y^2}\,. \tag{4.18}$$

Figure 4.12 shows with bars the infinite norms of the errors using the optimal variable shape parameters $\tilde{c}_i^+$ for the four problems (4.15)-(4.18). In these problems, we have used a regular mesh of 31×31 nodes. Also shown with dashed lines are the infinite norms of the errors with constant values of the shape parameter c.

In the top left image of Fig. 4.12 we show the results from problem (4.15). This is a very peculiar problem because an optimal value of the shape parameter $\tilde{c}_i^+$ exists for all nodes, and the value of $\tilde{c}_i^+$ is the same for all of them (notice that the bar for the optimal $\tilde{\mathbf{c}}^+$ is just one point). This is because the solution of this problem is an eigenfunction of the Laplacian. Since $\tilde{c}_i^+$ is independent of the node location x_i, the resulting error can be made as small as needed by just computing the value of c^* with sufficient accuracy. These results are also summarized in Table 4.6.

The top right image of Fig. 4.12 shows the results from problem (4.16).

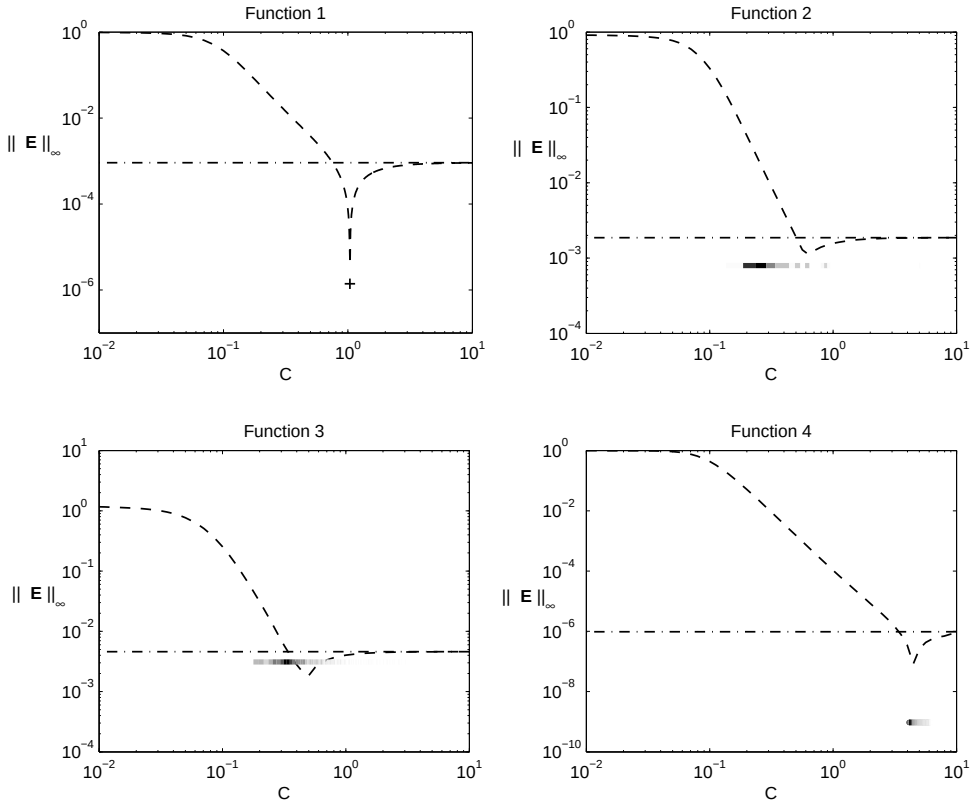


Figure 4.12: Infinite norm of the errors of problem (4.13) with exact solutions (4.15) to (4.18), using $N = 31 \times 31$ structured nodes. Dashed line; global RBF-FD error with constant c. Dot-dashed line; finite difference error. Bar: global RBF-FD error with optimal variable c (the gray scale in the bar is proportional to the number of nodes with a particular shape parameter).

Table 4.6: Same as Table 4.1 but for the Laplacian problem with solutions (4.15)-(4.18). In every case, N is equal to 31×31 structured nodes.

Structured nodes							
f	$\min(\tilde{\mathbf{c}}^+)$	$\max(\tilde{\mathbf{c}}^+)$	$\%N$	$\|\|\mathbf{E}(\tilde{\mathbf{c}}^+)\|\|_\infty$	$\tilde{c}^*$	$\|\|\mathbf{E}(\tilde{c}^*)\|\|_\infty$	$\|\|\mathbf{E}\|\|_\infty$
u_1	1.0387	1.0387	0	$1.394 \cdot 10^{-6}$	1.0387	$1.345 \cdot 10^{-6}$	$9.144 \cdot 10^{-4}$
u_2	0.2566	5.1882	74.1	$8.000 \cdot 10^{-4}$	0.5978	$1.043 \cdot 10^{-3}$	$1.868 \cdot 10^{-3}$
u_3	0.2500	7.8858	17.5	$3.032 \cdot 10^{-3}$	0.4935	$1.758 \cdot 10^{-3}$	$4.604 \cdot 10^{-3}$
u_4	4.1053	6.2108	0	$9.405 \cdot 10^{-10}$	4.4957	$7.440 \cdot 10^{-8}$	$9.727 \cdot 10^{-7}$

In this case, there is a very small improvement in accuracy. The reason for this can be explained by considering the results in Table 4.6. Notice that there is a high percentage of nodes ($\%N = 74.1\%$) for which $\tilde{c}_i^+$ does not exist and, therefore, in which the conventional FD approximation is used with the corresponding deterioration of the overall accuracy.

A similar behavior is observed for problem (4.17), where the accuracy is worse with the optimal variable shape parameter $\tilde{c}_i^+$ than with the optimal constant one c^* (see the bottom left image of Fig. 4.12).

On the contrary, problem (4.18) is an example where the use of a variable shape parameter leads to a very significant improvement of the accuracy (see the bottom right image of Fig. 4.12). In fact, the infinite norm of the error with the optimal variable $\tilde{\mathbf{c}}^+$ is $9.405 \cdot 10^{-10}$, nearly two orders of magnitude improvement with respect to the constant optimal value c^* for which the error is $7.440 \cdot 10^{-8}$. Again, the reason for this high accuracy is that, in this case, there is an optimal shape parameter $\tilde{c}_i^+$ for all nodes ($\%N = 0$ in the last row of Table 4.6).

To overcome the problem of the existence of optimal shape parameters $\tilde{c}_i^+$ for problems (4.16) and (4.17), we have solved them using the RBF (4.11) in a way analogous to that described in Section 4.5 for convection-diffusion problems. The corresponding local truncation error is

$$\begin{aligned} \tau_5(\mathbf{x}_i, c_i, \beta_i) &= \frac{h^2}{12}\left(u^{(4,0)}(\mathbf{x}_i) + u^{(0,4)}(\mathbf{x}_i)\right) \\ &- \frac{h^2}{c_i^2}\frac{(\beta_i - 6)(\beta_i - 2)}{2(\beta_i - 4)}\left(u^{(2,0)}(\mathbf{x}_i) + u^{(0,2)}(\mathbf{x}_i)\right) \qquad (4.19) \\ &+ \frac{h^2}{c_i^4}\frac{(\beta_i - 8)(\beta_i - 2)\beta_i}{2(\beta_i - 4)}\, u(\mathbf{x}_i) \; + \; O\left(h^3\, P_2(1/c_i^2)\right). \end{aligned}$$

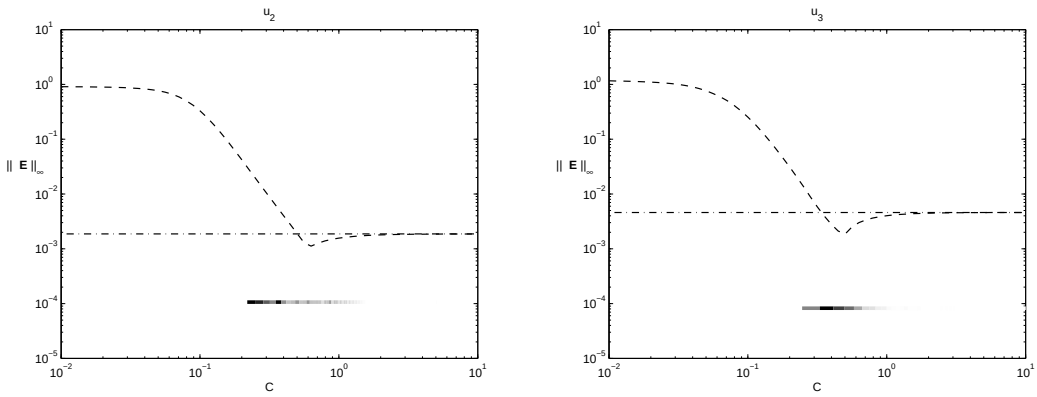


Figure 4.13: Infinite norms of the errors from problem (4.13) with exact solutions (3.21) and (3.23) using the generalized multiquadrics (4.11). As in Fig.4.12, we use $N = 31 \times 31$ structured nodes.

The new (node-dependent) parameter β_i^+ in (4.11) is chosen so that there exists a valid optimal shape parameter $\tilde{c}_i^+ \gg h$ at each node of the grid, as explained in Section 4.5. Figure 4.13 shows the results corresponding to problems (4.16) and (4.17) using the generalized multiquadric RBF (4.11) with $c_{min} = 0.2$. In both cases, there is an improvement of two orders of magnitude in the accuracy of the computed solutions.

4.7 Conclusions

In this chapter we present a novel technique to compute the solution of PDEs with the multiquadric RBF-FD method using an optimal variable shape parameter c_i at each node of the computational domain. We show that the numerical strategy can give rise to several orders more accurate solutions if there exists an optimal value of the shape parameter for most of the grid points of the domain. However, if there are many nodes for which an optimal value of c does not exist, the accuracy is similar to that obtained with an optimal constant c or even with standard FD formulas. For those cases, we notice that using generalized multiquadrics as RBFs and choosing the exponent β and the shape parameter c appropriately, guarantees the existence of an optimal shape parameter for all the nodes. In this way, we are able to obtain significant improvements in accuracy for all the examples analyzed both with structured and unstructured grids.

Chapter 5

Optimal Shape Parameter for the solution of Elastostatic Problems with the RBF Method

5.1 Outline

Finite Element Method (FEM) has become the standard method to numerically solve solid mechanics problems. It is very well suited for problems with irregular geometries which are solved using unstructured grids. However, during the last years considerable efforts have been devoted to *meshless* methods which operate with nodes instead of meshes. Their main advantage is that no mesh generation is required, thus eliminating one of the most complex steps in the solution procedure. Meshless methods are specially well suited for problems with large deformations, moving discontinuities or problems that require frequent remeshing.

Zhang *et al* [111] were the first authors to investigate the capabilities of the global RBF method for the solution of elasticity problems. They considered both globally supported RBFs, such as multiquadrics and thin-plate splines, and compactly supported RBFs, such as Wendland's functions [105]. Later, Tolstykh and Shirobokov [100] applied the local RBF method to the same elasticity problems analyzed in [111] (cantilever beam, plate with a circular hole). Both the global [111] and local [100] approaches are based on the strong formulation of the linear elasticity equations. A third alternative to use RBFs for elasticity problems was proposed by Liu *et al* [74, 103]. It combines the Galerkin weak form and RBFs to form a Radial Point Interpolation Method (RPIM). Its performance was analyzed by solving the cantilever beam and the plate with a circular hole problems.

During the last years, the RBF method has been successfully used to solve a large variety of solid mechanics problems using either the global method [30, 31, 33, 35, 36, 68, 85], or the local method [32, 34, 84], or RPIM [71, 75]. In most of these works multiquadrics have been used as RBF, and it is well known that accuracy is strongly dependent on the value of the shape parameter. For the local RBF method, we have proposed an efficient procedure to compute both, the optimal constant shape parameter (Chapter 3) and the optimal variable one (Chapter 4). The objective of this work is to apply these procedures to elasticity problems and show how the accuracy can be significantly increased by efficiently tuning the values of shape parameters.

This chapter is organized as follows: in Section 5.2 we describe the formulation of global and local RBF methods for the solution of plane stress problems. In Section 5.3 we apply these methods to the solution of the Timoshenko's beam problem and to the problem of an infinite plate with a circular hole. These problems are solved both with structured and unstructured nodes using either a constant or a variable shape parameter. Section 5.4 contains the main conclusions of this chapter.

5.2 Formulation

In the case of plane stress problems, the equations of elasticity written in terms of displacements are,

$$\frac{E}{1-\nu^2}\begin{bmatrix} \dfrac{\partial^2}{\partial x^2}+\dfrac{1-\nu}{2}\dfrac{\partial^2}{\partial y^2} & \dfrac{1+\nu}{2}\dfrac{\partial^2}{\partial x\partial y} \\ \dfrac{1+\nu}{2}\dfrac{\partial^2}{\partial x\partial y} & \dfrac{\partial^2}{\partial y^2}+\dfrac{1-\nu}{2}\dfrac{\partial^2}{\partial x^2} \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}=\begin{bmatrix} f_x \\ f_y \end{bmatrix}. \tag{5.1}$$

These equations have to be solved with appropriate boundary conditions. Once these equations are solved for the displacements, the corresponding stresses can be obtained through,

$$\sigma_{xx} = \frac{E}{1-\nu^2}\left(\frac{\partial u}{\partial x}+\nu\frac{\partial v}{\partial y}\right), \tag{5.2a}$$

$$\sigma_{yy} = \frac{E}{1-\nu^2}\left(\nu\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right), \tag{5.2b}$$

$$\tau_{xy} = \frac{E}{2(1+\nu)}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right). \tag{5.2c}$$

5.2.1 Global RBF Method

In the global RBF method (see Section 1.3), we look for an approximate solution in the space spanned by a set of translated RBFs. Thus,

$$u(\mathbf{x}) = \sum_{k=1}^{N} a_k \phi(||\mathbf{x} - \mathbf{x}_k||, c_u), \tag{5.3a}$$

$$v(\mathbf{x}) = \sum_{k=1}^{N} b_k \phi(||\mathbf{x} - \mathbf{x}_k||, c_v), \tag{5.3b}$$

where $||\cdot||$ is the Euclidean norm, $\mathbf{x}_k$ is a set of N RBF centers and ϕ is an RBF centered at $\mathbf{x}_k$. The unknown coefficients, a_k and b_k, are the coordinates of the approximate solution in the functional space spanned by the RBFs. There are a large variety of functions that can be used as RBFs (see Table 1.1). As in the previous chapters, we will use Hardy's multiquadric (1.3).

The unknown coefficients, a_k, b_k, are computed by collocation of equations (5.1) at a set of interior nodes and collocation of the boundary conditions at boundary nodes. For convenience we will use the same set of RBF centers as collocation nodes. Let's define a vector $\mathbf{z}$ of length $2N$ containing the unknowns,

$$\begin{cases} z_k = a_k \\ z_{k+N} = b_k \end{cases}, \qquad k = 1, \ldots, N,$$

and a vector $\mathbf{f}$ containing the forcing terms such that,

$$\begin{cases} f_k = f_x(\mathbf{x}_k) \\ f_{k+N} = f_y(\mathbf{x}_k) \end{cases}, \qquad k = 1, \ldots, N.$$

Substituting equations (5.3a-5.3b) in equation (5.1) leads to the following linear system,

$$A\mathbf{z} = \mathbf{f}, \tag{5.4}$$

where the elements of matrix A corresponding to an interior node k are, for

$i = 1, \ldots, N,$

$$\begin{aligned} A_{k,i} &= \frac{E}{1-\nu^2}\left[\frac{\partial^2\phi_k}{\partial x^2}(\mathbf{x}_i) + \frac{1-\nu}{2}\frac{\partial^2\phi_k}{\partial y^2}(\mathbf{x}_i)\right], \\ A_{k,i+N} &= \frac{E}{2(1-\nu)}\frac{\partial^2\phi_k}{\partial x\,\partial y}(\mathbf{x}_i), \\ A_{k+N,i} &= \frac{E}{2(1-\nu)}\frac{\partial^2\phi_k}{\partial x\,\partial y}(\mathbf{x}_i), \\ A_{k+N,i+N} &= \frac{E}{1-\nu^2}\left[\frac{\partial^2\phi_k}{\partial y^2}(\mathbf{x}_i) + \frac{1-\nu}{2}\frac{\partial^2\phi_k}{\partial x^2}(\mathbf{x}_i)\right]. \end{aligned}$$

For boundary collocation nodes the elements of matrix A depend on the boundary condition. For instance, in the case of Dirichlet boundary conditions,

$$A_{k,i} = \phi_k(||\mathbf{x}_i - \mathbf{x}_k||, c_u), \qquad f_k = u_e(\mathbf{x}_k),$$

and

$$A_{k+N,i+N} = \phi_k(||\mathbf{x}_i - \mathbf{x}_k||, c_v), \qquad f_{k+N} = v_e(\mathbf{x}_k).$$

Solution of equation (5.4) yields the vector $\mathbf{z}$ and therefore the coefficients a_k and b_k of equations (5.3a-5.3b) which are used to compute the horizontal and vertical displacements u and v. Also, the stresses are computed from (5.2a-5.2c) with

$$\frac{\partial u}{\partial x} = \sum_{k=1}^{N} z_k \frac{\partial\phi_k(||\mathbf{x} - \mathbf{x}_k||, c_u)}{\partial x}, \tag{5.6a}$$

$$\frac{\partial u}{\partial y} = \sum_{k=1}^{N} z_k \frac{\partial\phi_k(||\mathbf{x} - \mathbf{x}_k||, c_u)}{\partial y}, \tag{5.6b}$$

$$\frac{\partial v}{\partial x} = \sum_{k=1}^{N} z_{k+N} \frac{\partial\phi_k(||\mathbf{x} - \mathbf{x}_k||, c_v)}{\partial x}, \tag{5.6c}$$

$$\frac{\partial v}{\partial y} = \sum_{k=1}^{N} z_{k+N} \frac{\partial\phi_k(||\mathbf{x} - \mathbf{x}_k||, c_v)}{\partial y}. \tag{5.6d}$$

5.2.2 Local RBF Method

In the local RBF method (see Section 1.4), the action of the differential operator $\mathcal{L}[\cdot]$ on a function evaluated at a node $\mathbf{x}_i$ is approximated by the weighted

sum (1.17). For instance, the second derivative of horizontal displacement with respect to x, u_{xx}, appearing in equation (5.1) is approximated by

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathbf{x}=\mathbf{x}_i} \approx \sum_{j=1}^{n} \alpha_{ij} u(\mathbf{x}_j). \tag{5.7}$$

The coefficients α_{ij} are computed by imposing (5.7) to be exact for RBFs, what yields the linear system of equations (1.18).

The coefficients to approximate u_{yy}, u_{xy}, v_{xx}, v_{yy} and v_{xy} are similarly computed. Notice, that in principle there are 6 shape parameters that have to be chosen ($c_{uxx}, c_{uyy}, c_{uxy}, c_{vxx}, c_{vyy}, c_{vxy}$).

5.3 Numerical examples

5.3.1 Cantilever Beam

As a first example we consider a cantilever beam of depth D, length L and unit thickness, which is fully fixed to a support at $x = 0$ and carries an end load P. The analytic solution of this problem is given by Timoshenko and Goodier [98, chap 3] as,

$$u_e = -\frac{P}{6\,E\,I}\left(y - \frac{D}{2}\right)\left[(6\,L - 3\,x)\,x + (2+\nu)\,(y^2 - D\,y)\right], \tag{5.8}$$

$$v_e = \frac{P}{6\,E\,I}\left[3\,\nu\left(y - \frac{D}{2}\right)^2 (L-x) + (4+5\,\nu)\,\frac{D^2 x}{4} + (3\,L - x)\,x^2\right], \tag{5.9}$$

where E is Young's modulus, ν is Poisson's ratio and I is the second moment of area of the cross section ($I = D^3/12$ for a narrow rectangular beam of width 1). The stresses corresponding to the above displacements are,

$$\sigma_{xx} = -\frac{P\,(L-x)}{I}\left(y - \frac{D}{2}\right), \tag{5.10a}$$

$$\sigma_{yy} = 0, \tag{5.10b}$$

$$\tau_{xy} = -\frac{P}{2\,I}\left(y^2 - D\,y\right). \tag{5.10c}$$

This problem has been widely used to demonstrate the capabilities of adaptive procedures in FEMs, meshless methods and other numerical techniques. In many cases, however, the boundary conditions necessary to match the exact solution are not used, and therefore conclusions based on errors computed

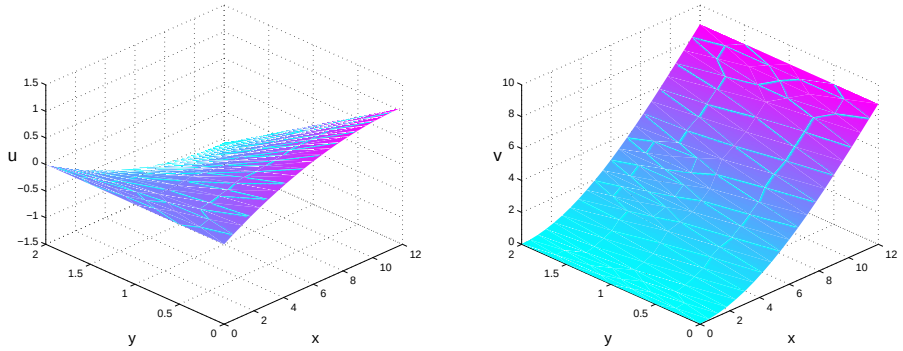


Figure 5.1: Exact solution of cantilever beam (5.8, 5.9) for the case $L = 12$, $D = 2$, $E = 1000$, $\nu = 0.3$. Left: horizontal displacement u. Right: vertical displacement v.

using those solutions are questionable. This fact, has been recently pointed out by Augarde and Deeks [1]. These authors remark that the displacements given in (5.8-5.9) are an exact solution of the plane stress equations only if the load is distributed parabolically (as in 5.10c) and if essential boundary conditions are applied at $x = 0$ according to equations (5.8-5.9).

To be specific, let's consider the displacements of a beam of length $L = 12$, width $D = 2$, Young modulus $E = 1000$ and Poisson ratio $\nu = 0.3$, to which a vertical force $P = 10$ per unit length is applied at its free end.

The solution given by (5.8-5.9) is shown in Figure 5.1. According to [1] this is the exact solution of the problem with free boundary conditions at $y = 0$ and $y = D$, Dirichlet boundary conditions at $x = 0$ given by (5.8-5.9), and parabolic load at $y = L$,

$$P' = -\frac{6\,P}{D^3}\,(y^2 - D\,y), \qquad \int_0^D P'\,dy = P.$$

However, the Timoshenko beam problem is often solved with other boundary conditions. For instance, a common boundary condition used is full-fixity at the support, and uniform load P at the vertical surface at $x = L$. The solution in terms of displacements with these boundary conditions is very similar to the solution shown in Figure 5.1. However, the solution in terms of stresses is quite different as can be observed in Figures 5.2 and 5.3. These results have been obtained using FEMs with a mesh of 6985 nodes and 13568 triangular elements. Figure 5.2 compares the distribution of σ_{xx} of the exact solution (5.10c) (left part), with the corresponding distribution of σ_{xx} for the problem with fully fixed boundary conditions at $x = 0$ ($u = v = 0$) and

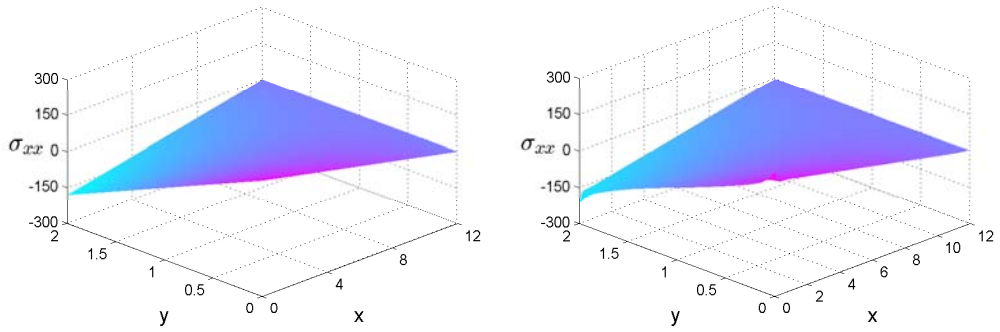


Figure 5.2: x stress in the case $L = 12$, $D = 2$, $E = 1000$, $\nu = 0.3$. Left: σ_{xx} (5.10a) distribution for exact solution. Right: σ_{xx} distribution for boundary conditions $u = v = 0$ at $x = 0$ and $P' = 5$ per unit length at $x = L$.

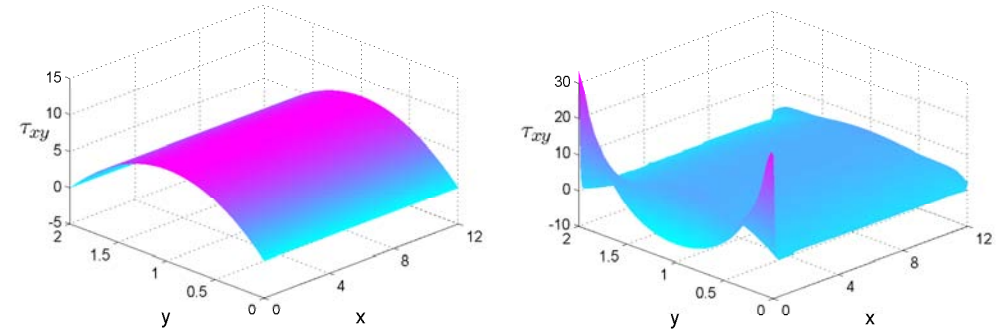


Figure 5.3: Shear stress in the case $L = 12$, $D = 2$, $E = 1000$, $\nu = 0.3$. Left: τ_{xy} (5.10c) distribution for exact solution. Right: shear stress τ_{xy} distribution for boundary conditions $u = v = 0$ at $x = 0$ and $P' = 5$ per unit length at $x = L$.

uniform load $P' = 5$ per unit length at $x = L$ (right part). Notice that both solutions are quite similar except near $x = 0$ where they differ significantly due to stress concentrations near the borders.

Showing even greater differences are the results for shear stress shown in Figure 5.3. Notice that the shear stress distribution exhibits singularities at the top and bottom corners which lead to significant differences with the Timoshenko beam solution (5.10a-5.10c) throughout the beam.

Other boundary conditions frequently used for the Timoshenko beam problem are:

(A) $u = v = 0$ at $x = 0$ and $P' = 5$ per unit length at $x = L$ (Figures 5.2 and 5.3).

(B) $u = v = 0$ at $x = 0$ and $P' = -(6\,P)/D^3)\,(y^2 - D\,y)$ at $x = L$.

(C) $u = 0$ at $x = 0, u = v = 0$ at $x = 0, y = D/2$, and $P' = -(6\,P)/D^3)\,(y^2 - D\,y)$ at $x = L$.

(D) $P' = (6\,P)/D^3)\,(y^2 - D\,y)$ at $x = 0$ and $P' = -(6\,P)/D^3)\,(y^2 - D\,y)$ at $x = L$.

(E) u and v given by (5.8-5.9) at $x = 0$ and $P' = -(6\,P)/D^3)\,(y^2 - D\,y)$ at $x = L$.

All these boundary conditions except (E) lead to solutions slightly different from (5.8-5.9). In the rest of the paper we will use Dirichlet boundary conditions given by (5.8-5.9) in all boundaries of the beam $x = 0$, $x = L$, $y = 0$, $y = D$ and in this way we will be able to use the exact solution for displacements (5.8-5.9) and stresses (5.10a-5.10c) when computing numerical errors with the proposed RBF methods.

5.3.1.1 FEM solution

For comparison purposes, we will first compute the solution using FEMs on a mesh of 132 nodes and 212 elements. We denote the solution in horizontal and vertical displacements as u_{FEM} and v_{FEM} and the corresponding errors as,

$$\bar{\mathbf{E}}_u = u_{\text{FEM}} - u_e \qquad \text{and} \qquad \bar{\mathbf{E}}_v = v_{\text{FEM}} - v_e.$$

Figure (5.4) shows the error in horizontal and vertical displacements ($\bar{\mathbf{E}}_u$ and $\bar{\mathbf{E}}_v$) resulting from solving the Timoshenko beam problem with Dirichlet boundary conditions using FEM. The infinity norm of the error in u is

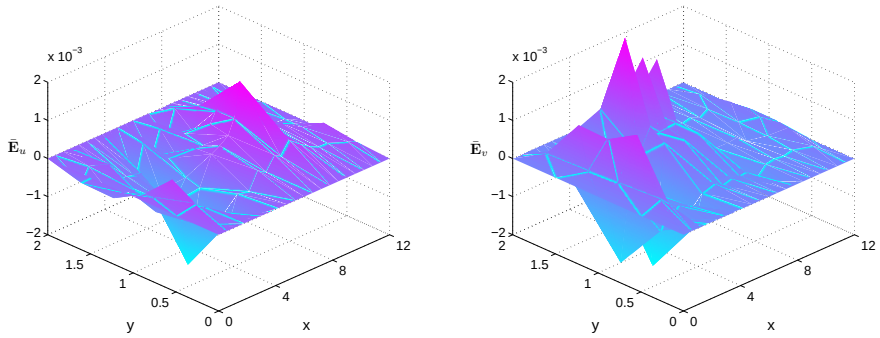


Figure 5.4: Error of FEM solution with 132 nodes and 212 elements. Left: error distribution for horizontal displacement $\bar{\mathbf{E}}_u$. Right: error distribution for vertical displacement $\bar{\mathbf{E}}_v$.

$||\bar{\mathbf{E}}_u||_\infty = 2.544 \times 10^{-3}$ and the Euclidean norm $||\bar{\mathbf{E}}_u||_2 = 5.147 \times 10^{-3}$. The corresponding values for vertical displacements are $||\bar{\mathbf{E}}_u||_\infty = 2.881 \times 10^{-3}$ and $||\bar{\mathbf{E}}_u||_2 = 6.931 \times 10^{-3}$.

5.3.1.2 Global RBF solution

We can compute the solution with the global RBF method using as RBF centers exactly the same nodes of the FEM mesh. For convenience we will use the same value of the shape parameter both for horizontal, u, and vertical, v, displacements ($c_u = c_v = c$). We will use the following measure of error,

$$\mathbf{E}_u = u - u_e, \qquad \text{and} \qquad \mathbf{E}_v = v - v_e,$$

so that,

$$E = ||\mathbf{E}_u||_\infty + ||\mathbf{E}_v||_\infty. \tag{5.11}$$

Figure 5.5 shows the sum of the infinity norms of the errors in horizontal and vertical displacements of the global RBF solution as a function of the shape parameter c. The solid line shows the results obtained with the same $N = 132$ unequally spaced nodes used in the FEM solution, and the dashed line the results obtained on an equally spaced mesh of $N = 19 \times 7 = 133$ equally spaced nodes. Notice that both curves exhibit the exponential convergence of the error with increasing c. Also notice that for values of shape parameter larger than approximately 6 the resulting matrix becomes ill-conditioned and convergence is no longer a smooth function. It should be pointed out that for the same number of nodes the global RBF solution is significantly more accurate than the FEM solution. For instance, the results

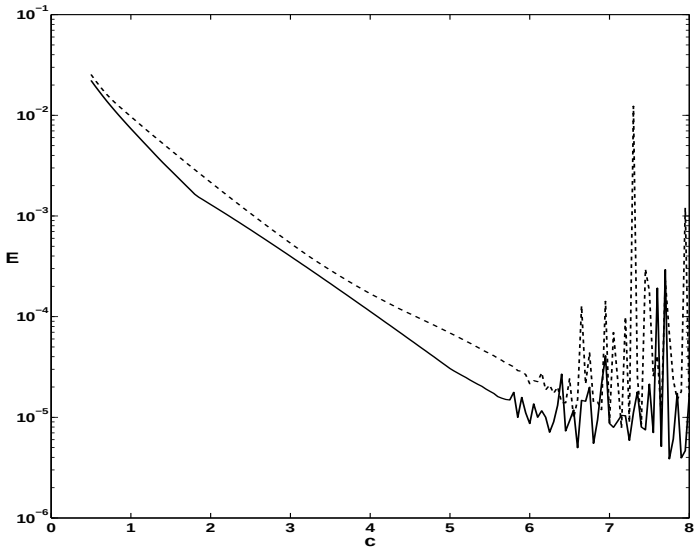


Figure 5.5: Error of global RBF solution, E, as a function of the shape parameter c. Solid: $N = 132$ non-equispaced RBF centers. Dashed: $N = 19 \times 7 = 163$ equispaced nodes.

for a value of the shape parameter $c = 6$ are shown in Figure 5.6. Notice, that there is approximately three orders of magnitude increase in accuracy if one compares Figure 5.6 to the error of the FEM solution shown in Figure 5.4. In fact, the infinity norm of the error in u is $||\mathbf{E}_u||_\infty = 1.868 \times 10^{-5}$ and the Euclidean norm $||\mathbf{E}_u||_2 = 6.348 \times 10^{-5}$. The infinity norm of the error in v is $||\mathbf{E}_v||_\infty = 2.872 \times 10^{-6}$ and the Euclidean norm $||\mathbf{E}_v||_2 = 1.027 \times 10^{-5}$. However, it should be pointed out that the linear system which has to be solved in order to compute the numerical solution with the global RBF method is dense, while in the case of the FEM is sparse. Thus, for the same number of nodes, the computational cost of the global RBF method is higher than the computational cost of the FEM.

5.3.1.3 Local RBF solution

To compute the solution with the RBF local method we have to select a set of N nodes and for each node a stencil of n surrounding nodes. For convenience we will use a grid of equispaced nodes and, for each node (x_i, y_i) a stencil of three adjacent nodes (x_i, y_i), $(x_i - \Delta x, y_i)$, $(x_i + \Delta x, y_i)$ in the horizontal direction to approximate ∂_{xx}, three nodes in the vertical direction

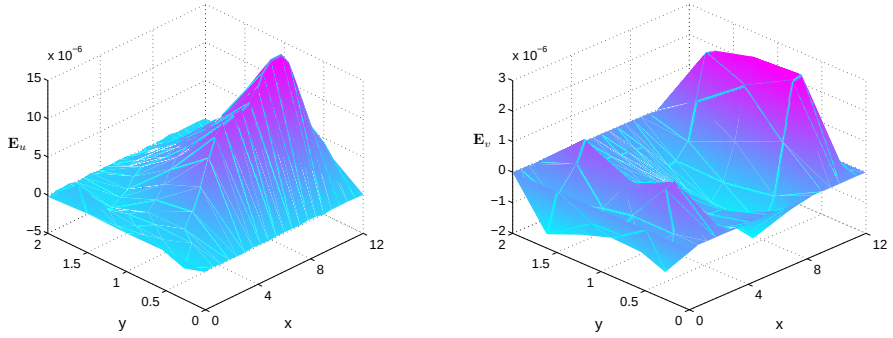


Figure 5.6: Error of global RBF solution for $c_u = c_v = 6$ and $N = 132$ RBF centers. Left: error distribution for horizontal displacement $\mathbf{E}_u$. Right: error distribution for vertical displacement $\mathbf{E}_v$.

(x_i, y_i), $(x_i, y_i - \Delta y)$, $(x_i, y_i + \Delta y)$ to approximate ∂_{yy}, and five nodes (x_i, y_i), $(x_i - \Delta x, y_i - \Delta y)$, $(x_i + \Delta x, y_i - \Delta y)$, $(x_i + \Delta x, y_i + \Delta y)$, $(x_i - \Delta x, y_i + \Delta y)$ to approximate ∂_{xy}.

Figure 5.7 shows the error as a function of the shape parameter for a grid of $N = 19 \times 7 = 133$ equispaced nodes. In this case we have taken the shape parameters for all the derivatives appearing in (5.1) as equal ($c_{uxx} = c_{uyy} = c_{uxy} = c_{vxx} = c_{vyy} = c_{vxy} = c$). Notice that the exponential convergence of the global method has been lost and that the error E of the local method decreases as c^{-2}. This behavior should be expected according to the formulas for the error of the RBF local method derived in Chapter 2. In fact, the formula for the error in approximating the second derivative with respect to x using three equispaced nodes (2.11) is

$$\tau_3(u_{xx}) = \frac{(\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4} + \frac{(\Delta x)^2}{c^2}\frac{\partial^2 u}{\partial x^2} - \frac{3\,(\Delta x)^2}{4\,c^4}u + O\left((\Delta x)^4\right).$$

Similarly,

$$\tau_3(u_{yy}) = \frac{(\Delta y)^2}{12}\frac{\partial^4 u}{\partial y^4} + \frac{(\Delta y)^2}{c^2}\frac{\partial^2 u}{\partial y^2} - \frac{3(\Delta y)^2}{4c^4}u + O\left((\Delta y)^4\right).$$

Since the solutions for u and v of the Timoshenko's beam problem (5.8-5.9) are polynomials of third degree, then u_{xxxx}, u_{yyyy}, v_{xxxx} and v_{yyyy} are zero. Thereby, the errors in approximating the second derivatives of u and v approach zero as c approaches infinity. In fact, it is well known that RBF-FD formulas approach standard FD when $c \to \infty$. For this particular problem, standard FD yield the exact solution of the problem, and the RBF-FD error approaches zero as $c \to \infty$ ($E \approx O(c^{-2})$).

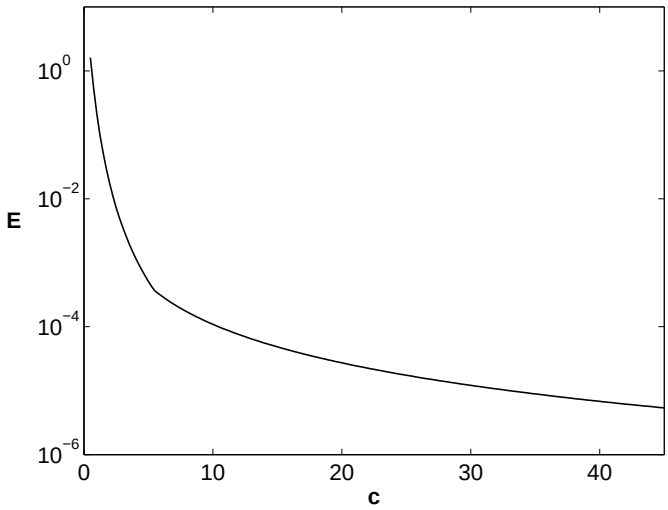


Figure 5.7: Error of RBF local solution for as a function of c. Equispaced grid of $N = 19 \times 7 = 133$ nodes.

5.3.2 Plate with a Hole

Let's consider the problem of an infinite plate with a hole of radius a loaded by a traction σ_0 at infinity in the x direction [98]. This problem has been often used as a test case to assess the accuracy of different meshless methods [15, 100, 111]. In Cartesian coordinates the exact solution can be written,

$$u_e = \frac{\sigma_0 a}{8G}\left[\frac{x}{a}(\kappa+1) + 2a(1+\kappa)\frac{x}{x^2+y^2} + 2a\frac{x^3-3xy^2}{(x^2+y^2)^2}\left(1-\frac{a^2}{x^2+y^2}\right)\right], \quad (5.12)$$

$$v_e = \frac{\sigma_0 a}{8G}\left[\frac{y}{a}(\kappa-3) + 2a(1-\kappa)\frac{y}{x^2+y^2} + 2a\frac{3yx^2-y^3}{(x^2+y^2)^2}\left(1-\frac{a^2}{x^2+y^2}\right)\right], \quad (5.13)$$

where

$$G = \frac{E}{2\,(1+\nu)} \qquad \text{and} \qquad \kappa = \frac{3-\nu}{1+\nu}\,.$$

The corresponding stresses are,

$$\begin{aligned}
\sigma_{xx} &= \sigma_0\left[1 - \tfrac{3}{2}\frac{a^2(x^2-y^2)}{(x^2+y^2)^2} - \frac{a^2}{(x^2+y^2)^3}(x^4+y^4-6x^2y^2)\left(1-\tfrac{3}{2}\frac{a^2}{x^2+y^2}\right)\right],\\
\sigma_{yy} &= \sigma_0\left[-\tfrac{1}{2}\frac{a^2(x^2-y^2)}{(x^2+y^2)^2} + \frac{a^2}{(x^2+y^2)^3}(x^4+y^4-6x^2y^2)\left(1+\tfrac{3}{2}\frac{a^2}{x^2+y^2}\right)\right],\\
\tau_{xy} &= -\sigma_0\left[\frac{a^2xy}{(x^2+y^2)^2} + 4a^2xy\frac{x^2-y^2}{(x^2+y^2)^3}\left(1+\tfrac{3}{2}\frac{a^2}{x^2+y^2}\right)\right].
\end{aligned}$$

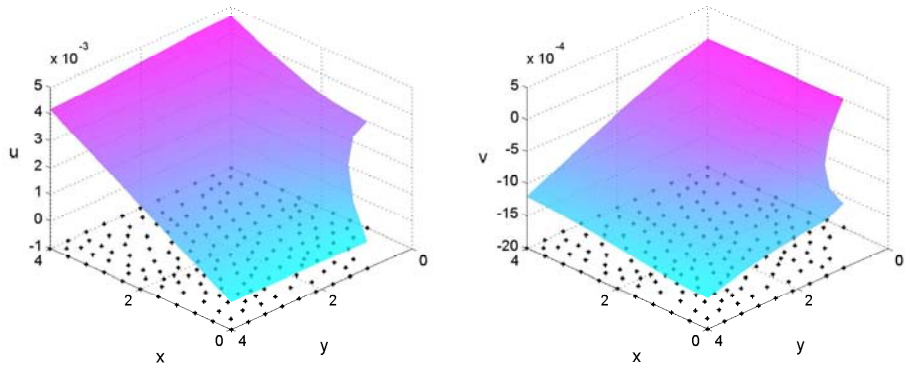


Figure 5.8: RBF global solution for the case $N = 174$, $c_u = c_v = 2$. Left: horizontal displacement u. Right: vertical displacement v.

5.3.2.1 Global RBF solution

We solve the problem with $\sigma_0 = 1$, $E = 1000$, $L_x = 4$, $L_y = 4$, $a = 1$, $\nu = 0.3$ and Dirichlet boundary conditions. We use a set of 174 RBF centers which coincide with the nodes of a triangular mesh used to compute the solution with the FEM. Figure 5.8 shows the RBF numerical solution for horizontal (left) and vertical (right) displacements using $c_u = c_v = 2$. Also shown are the nodes used as RBF centers.

Figure 5.9 shows the error (5.11) of the global RBF solution as a function of the shape parameter ($c_u = c_v = c$) for two different sets of RBF centers. In the case $N = 633$, the error exhibits exponential convergence until a value of $c \approx 1.6$ is reached for which the matrix becomes ill-conditioned and roundoff errors deteriorate the accuracy of the solution. The exponential convergence can also be observed in the case $N = 174$. However, in this case, ill-conditioning occurs for values of the shape parameter larger than those shown in the figure.

It is also possible to use a minimization routine (we have used function `fminsearch` of MATLAB) to find values of c_u and c_v that minimize the error. For the coarser grid ($N = 174$) the error E is minimum for $c_u = 2.8803$, $c_v = 1.5480$ ($||\mathbf{E}_u||_\infty = 1.72 \times 10^{-5}$, $||\mathbf{E}_v||_\infty = 1.26 \times 10^{-5}$). For the finer grid ($N = 653$) the error is minimum for $c_u = 1.6364$, $c_v = 1.4401$ ($||\mathbf{E}_u||_\infty = 1.21 \times 10^{-6}$, $||\mathbf{E}_v||_\infty = 9.10 \times 10^{-7}$).

Figure 5.10 shows the stress σ_{xx} along the left boundary ($x = 0$) for two sets of RBF centers. For $N = 174$ RBF centers there are significant discrepancies with the analytical solution, specially in the vicinity of the hole ($y \approx 1$). For $N = 653$ the solution is very accurate for all values of y.

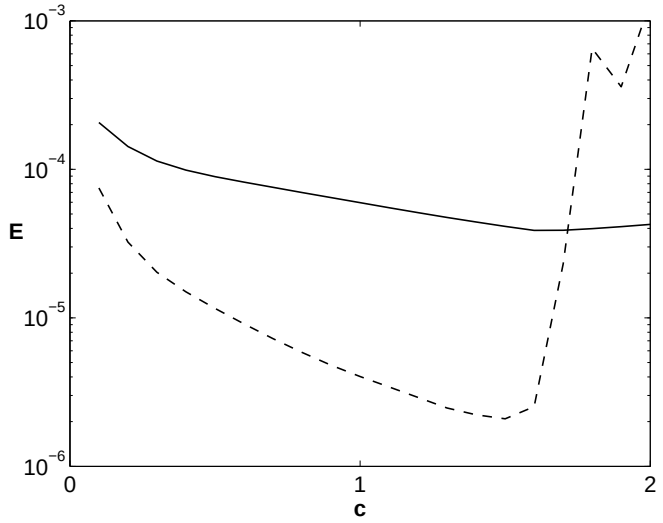


Figure 5.9: Error (5.11) as a function of the shape parameter c. Solid line: $N = 174$ RBF centers. Dashed line: $N = 653$ RBF centers.

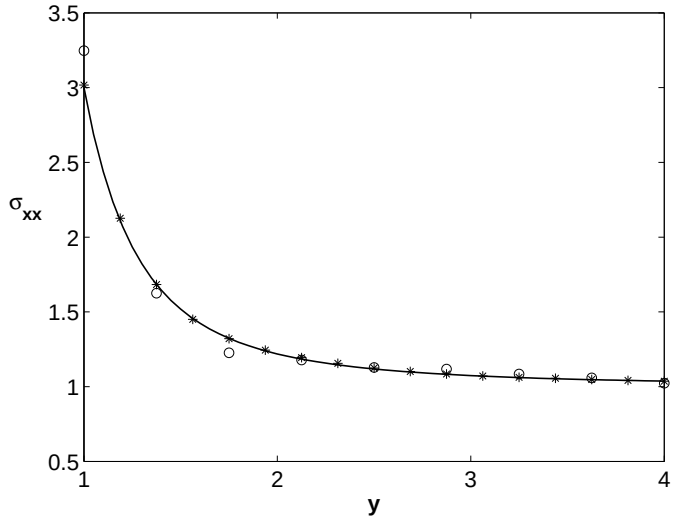


Figure 5.10: Stress σ_{xx} as a function of y at the left boundary ($x = 0$) obtained with the global RBF method. Solid line: exact solution. $\circ$: 174 RBF centers. $*$: 653 RBF centers.

5.3.2.2 Local RBF solution

The solution of the plate with a hole problem (5.12-5.13) is no longer a polynomial and therefore its solution with standard FD is no longer exact. We will use RBF-FD formulas to compute the solution both on an equispaced grid and on a non equispaced grid.

Equispaced nodes

For simplicity, we will start by using an equispaced grid made up by nodes which are in the boundary or in the interior of the plate. To compute second derivatives we use three node stencils; (x_i, y_i), $(x_i - \Delta x, y_i)$, $(x_i + \Delta x, y_i)$ to approximate u_{xx} and v_{xx}, and (x_i, y_i), $(x_i, y_i - \Delta y)$, $(x_i, y_i + \Delta y)$ to approximate u_{yy} and v_{yy}. To compute crossed derivatives we use five node stencils; (x_i, y_i), $(x_i - \Delta x, y_i - \Delta y)$, $(x_i - \Delta x, y_i + \Delta y)$, $(x_i + \Delta x, y_i - \Delta y)$, $(x_i + \Delta x, y_i + \Delta y)$ to approximate u_{xy} and v_{xy}. We consider as interior nodes those whose 6 neighbors belong to the set of nodes, and as boundary nodes those who have at least one neighbor that does not belong to the set of nodes. At interior nodes we apply the RBF-FD formulation and at boundary nodes we apply Dirichlet boundary conditions given by (5.12-5.13).

Figure 5.11 shows the error as a function of the shape parameter for three different grids: 21×21 ($N = 419$), 31×31 ($N = 909$), and 41×41 ($N = 1595$). In all cases the error increases for very small values of c, then decreases exponentially with increasing c and approaches a constant for larger values of c. This constant corresponds to the error resulting from standard FD.

In Chapter 2 we showed that frequently there is an optimal value of the shape parameter such that the local approximation error resulting from RBF-FD formulas is minimum. We also showed in Chapters 3 and 4 that using at each node the corresponding optimal shape parameter may lead to very significant improvements in accuracy. In the following we will apply this technique to compute accurate solutions of the plate with a hole problem.

For simplicity of the resulting formulas, we add a constant term to the RBF interpolation space [109], so that the RBF interpolant considered is

$$w(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \phi(r_i(\mathbf{x}), c) + \beta.$$

In Chapter 2 we derived an approximate equation for the local truncation error resulting from MQ RBF-FD formulas for second derivatives using three equispaced nodes (2.11). Adding the constant term, the error formula for u_{xx}

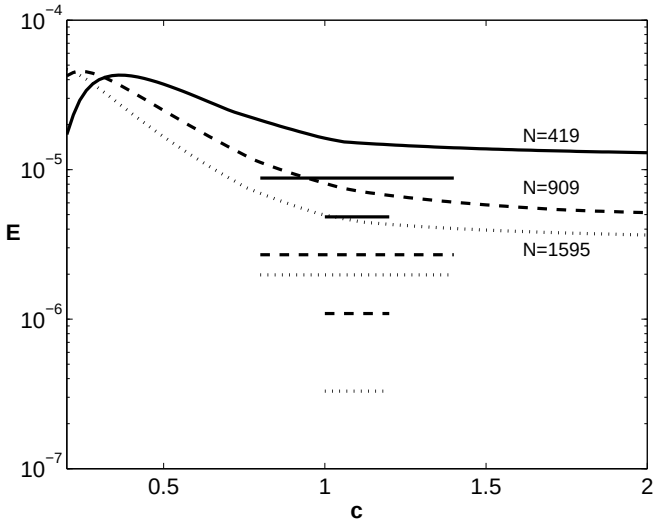


Figure 5.11: Error (5.11) as a function of the shape parameter c with the local method. Solid line: $N = 419$ RBF centers. Dashed line: $N = 909$ RBF centers. Dotted line: $N = 1595$ RBF centers. Long horizontal lines: Error (5.11) using the optimal, node-dependent shape parameter. Short horizontal lines: Error (5.11) using generalized multiquadrics.

is modified to,

$$\hat{\tau}_3(x,y) = \frac{(\Delta x)^2}{12}\frac{\partial^4 u}{\partial x^4} + \frac{5}{4}\frac{(\Delta x)^2}{c^2}\frac{\partial^2 u}{\partial x^2} + O\left((\Delta x)^4\right). \tag{5.14}$$

Therefore, the error is zero for

$$c^* = \sqrt{-15\,\frac{u_{xx}}{u_{xxxx}}}.$$

This is the optimal value that we use at each node to approximate second derivatives. Notice that, if the expression inside the square root is not positive, there is not any real value of c that makes the local approximation error null.

Analogously, for the cross derivative, the local approximation error using an equispaced five node stencil is,

$$\hat{\tau}_5(x,y) = \frac{1}{6}\left[\frac{\partial^4 u}{\partial x^3 \partial y}(\Delta x)^2 + \frac{\partial^4 u}{\partial x \partial y^3}(\Delta y)^2\right] + \frac{3}{2c^2}\frac{\partial^2 u}{\partial x \partial y}\left[(\Delta x)^2 + (\Delta y)^2\right] + O\left((\Delta x)^4, (\Delta y)^4\right). \tag{5.15}$$

In the case $\Delta x = \Delta y$, the optimal shape parameter is,

$$c^* = \sqrt{-\frac{18 u_{xy}}{u_{xxxy} + u_{xyyy}}}. \tag{5.16}$$

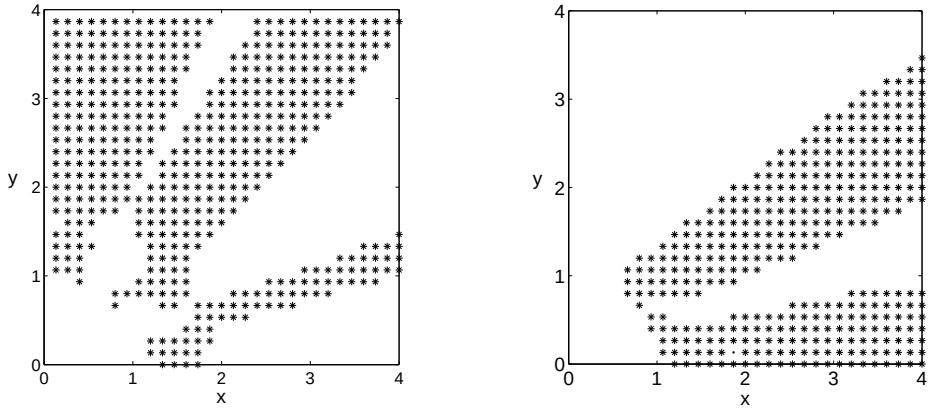


Figure 5.12: Left: nodes where optimal c to approximate u_{xx} exists. Right: nodes where optimal c to approximate u_{yy} exists.

Applying these formulas with the derivatives appearing in them computed from the exact solution (5.12, 5.13) at each node of the grid, we can compute the optimal shape parameter to approximate each of the derivatives of u and v appearing in (5.1). In real applications, when the exact solution is not known, the derivatives appearing in (5.16) are estimated using the procedure described in Chapter 3.

Figure 5.12 shows the nodes for which an optimal shape parameter exists for u_{xx} (left) and u_{yy} (right). These results correspond to the grid of 31×31 ($N = 909$) nodes. Notice that there are a large number of nodes for which no optimal shape parameter exists. In those nodes we use standard FD. Similar results apply for the optimal shape parameter corresponding to v_{xx}, v_{yy}, u_{xy} and v_{xy}.

Figure (5.13) shows the spatial distribution of optimal shape parameter for the cross derivative of the vertical displacement, v_{xy}. The results correspond to a grid of 41×41 ($N = 1595$) points. At nodes where no c^* exists we have assigned a value $c^* = 6$. For such a large value of c the RBF-FD formulas are equivalent to standard FD. Notice that the optimal shape parameter varies smoothly and its minimum value is 0.1810.

The error measures (5.11) resulting from applying the RBF-FD method with the optimal shape parameter at each node are shown as long horizontal lines in Figure 5.11 for the three grids analyzed. Notice that there are significant accuracy improvements with respect to the results obtained with a constant shape parameter for all nodes and all derivatives.

In Chapter 4 it was observed that the increase of accuracy resulting from the use of an optimal shape parameter at each node, strongly depended on

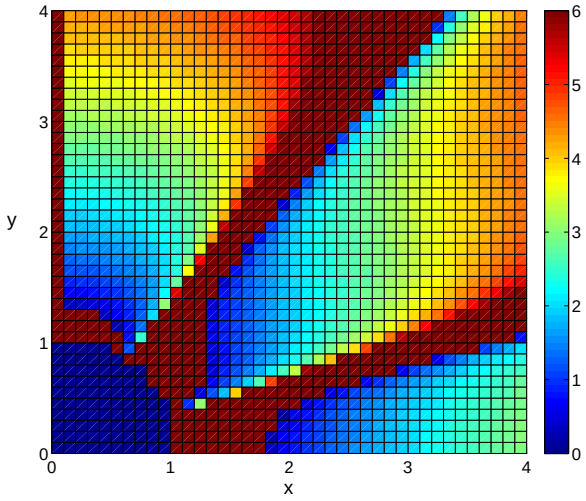


Figure 5.13: Distribution of optimal shape parameter corresponding to v_{xy}. $N = 1595$.

the percentage of nodes for which the optimal shape parameter exists. Hence, we proposed the use of generalized multiquadrics to ensure that the RBF-FD local truncation error is zero to leading order at all nodes. In fact, the generalized multiquadric (4.11) has an additional parameter β (node-dependent) which can be chosen so that an optimal value of the shape parameter exists at every node. Implementing this procedure to the solution of problem (5.12-5.13) results in the errors shown with a short horizontal line in Figure 5.11. As expected, this procedure leads to significant improvements in accuracy. For instance, in the case of $N = 1595$ the error (5.11) with FD is 3.94×10^{-6}, the error with the optimal shape parameter using standard multiquadrics is 1.97×10^{-6}, and the error with the optimal shape parameter using generalized multiquadrics is 4.52×10^{-7}. Similar improvements are observed for $N = 419$ and 909.

Non-equispaced nodes

We can also solve the problem with the local method using non-equispaced nodes. For each node, we select a stencil made up of the five nearest neighbors (stencils of 6 nodes), and for each differential operator and each node, we compute the coefficients α_{ij} of the stencil by solving the corresponding 6×6 linear system. Figure 5.14 shows the error (5.11) as a function of the shape parameter c for the grid of $N = 174$ nodes shown in Figure 5.8 (solid line).

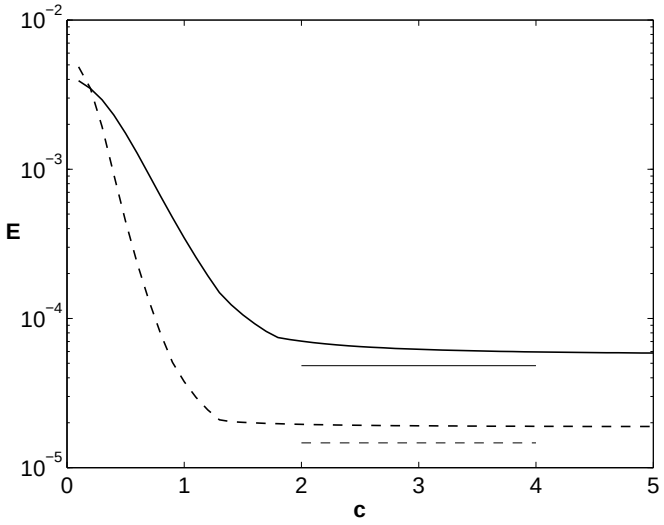


Figure 5.14: Error (5.11) as a function of the shape parameter c with the local method. Solid line: $N = 174$ RBF centers. Dashed line: $N = 653$ RBF centers. Thin lines: different shape parameters for each derivative.

Also shown is the error as a function of c for a grid of $N = 653$ nodes (dashed line) obtained by refining the initial mesh. Notice that the error dependence with c of the non-equispaced case is similar to that observed for the equispaced case (see Figure 5.11): the error decreases with increasing c and approaches a constant for large values of c. However, the errors for the non-equispaced case are significantly smaller than those of the equispaced case. The reason for this is that the nodes in the vicinity of the inner circle, where the errors are larger, are better distributed in the non-equispaced case that in the case of a regular equispaced grid. Also shown (thin lines) are the errors resulting from using a different shape parameter for each derivative. In the case $N = 175$ the optimal shape parameters for u_{xx}, u_{yy}, v_{xx}, v_{yy}, u_{xy} and v_{xy} are 14.2323, 1.7069, 1.4298, 1.9374, 5.3884 and 13.0723 respectively. In the case $N = 653$ the corresponding values of the shape parameters are 8.6361, 1.5943, 5.0336, 1.2891, 9.9104 and 8.1346 respectively. These values are obtained using function `fminsearch` of MATLAB to minimize the error. Notice that there is a slight improvement in accuracy.

5.4 Conclusions

In this chapter we analyze the applicability of the global and local versions of the RBF method to the solution of two standard elastostatic problems: the Timoshenko's beam problem and the problem of an infinite plate with a circular hole. The objective of our work is to show how the accuracy can be significantly increased by efficiently tuning the values of shape parameters. We use multiquadrics as RBFs both for the global and the local method, and describe how to select an optimal value of the shape parameter to minimize approximation errors. For the local method, the selection of the optimal shape parameter is based on analytical approximations to the local error derived in Chapter 2 using either the same shape parameter at all nodes as it is described in Chapter 3, or a node dependent shape parameter as it is described in Chapter 4. We use both equispaced and non-equispaced nodes and show that significant gains in accuracy result from a proper selection of the shape parameter.

Chapter 6

Propagation of premixed laminar flames in 3D narrow open ducts using RBF-generated finite differences

6.1 Introduction

Premixed flame propagation is an important topic in combustion research with many applications in engineering and industry safety. It arises in all occasions that a mixture of fuel and air occurs, and therefore the mixture can be ignited leading to a propagating flame. Thus, it is important to understand this physical process which very often takes place in complex shaped domains. One important tool to achieve this goal is the numerical simulation of the equations describing flame propagation. In the past, finite differences (FD) [10, 60, 61, 63, 81] and finite element methods (FEM) [47, 82] have been successfully used.

In this chapter we explore the feasibility of the RBF-FD method for the solution of flame propagation problems. The method has the great advantage that the resulting differentiation matrices are sparse and well-conditioned even for large scale problems in complexly shaped domains [38, 40, 88, 109]. We have considered a three-dimensional model for the propagation of flames in open circular ducts with a simple Arrhenius reaction term to model the chemistry. However, complex chemistry models can be easily incorporated since the RBF-FD method applies only to the modeling of the space derivative terms. We use stencils with a relatively large number of nodes what allows to achieve high order approximation accuracy. This approach was

first used in [5, 38, 40, 88], where it is proposed the use of RBF-FD methods with large stencils, achieving high accuracy and well-conditioning collocation systems.

The chapter is organized as follows: Section 6.2 describes the mathematical model; Section 6.3 explains the numerical implementation of the model using RBF-FD; Section 6.4 presents the two- and three-dimensional numerical results and Section 6.5 summarizes the main conclusions of the present work.

6.2 Mathematical model

The mathematical model which describes the propagation of laminar flames in open ducts of circular cross-section [60], written in the reference frame moving with a speed $V(t)$ relative to the solid wall, takes the form

$$\frac{\partial T}{\partial t} + V(t)\frac{\partial T}{\partial z} = \Delta T + \omega\left(T, Y\right), \tag{6.1a}$$

$$\frac{\partial Y}{\partial t} + V(t)\frac{\partial Y}{\partial z} = \frac{1}{Le}\Delta Y - \omega\left(T, Y\right), \tag{6.1b}$$

where Δ is the laplacian, Y denotes the mass fraction of reactant, T the non-dimensional temperature, z the longitudinal coordinate along the duct and $\omega\left(T, Y\right)$ the non-dimensional reaction rate, which is assumed to follow an Arrhenius law of the form

$$\omega\left(T, Y\right) = \frac{Ze^2}{2\,Le\;{u_p}^2} Y \exp\left[\frac{Ze\left(T-1\right)}{1+\gamma\left(T-1\right)}\right]. \tag{6.2}$$

The non-dimensional parameters Le, Ze and γ are the Lewis number, the Zeldovich number and the heat release ratio, respectively. u_p is the non-dimensional quantity

$$u_p = S_L/U_L = 1 - \frac{3\gamma + Le - 2.344}{Ze},$$

based on laminar burning velocity of planar flame [66]. In the following simulations, $Ze = 15$ and $\gamma = 0.8$ for different values of the Lewis number Le and the duct radius R. Two different cases are considered: an adiabatic wall,

$$\left.\frac{\partial Y}{\partial r}\right|_{r=R} = 0, \qquad \left.\frac{\partial T}{\partial r}\right|_{r=R} = 0,$$

and an isothermal wall,

$$\left.\frac{\partial Y}{\partial r}\right|_{r=R} = 0, \qquad T|_{r=R} = 0.$$

In both cases,

$$\begin{aligned} Y &= 1, & T &= 0; & x &\to -\infty, \\ \frac{\partial Y}{\partial z} &= 0, & \frac{\partial T}{\partial z} &= 0; & x &\to +\infty, \end{aligned}$$

far upstream and downstream of the flame front, respectively. The temperature T and concentration Y are assumed to be 2π-periodic functions of the azimuthal coordinate ϕ.

As initial conditions we have used

$$Y(z) = \frac{1}{1 + e^{25(z-z_0)}} \qquad \text{and} \qquad T(r,\phi,z) = [1 - Y(z)]\, f(r,\phi) \tag{6.3}$$

in the adiabatic case;

$$Y(z) = \frac{1}{1 + e^{25(z-z_0)}} \qquad \text{and} \qquad T(r,\phi,z) = \left[\frac{1 - Y(z)}{1 + e^{25(r-0.8R)}}\right] f(r,\phi) \tag{6.4}$$

in the isothermal case, where z_0 is the initial location of the flame and $f(r,\phi)$ is a function which modules the amplitude of the initial condition.

6.3 Numerical implementation

6.3.1 Spatial discretization

The spatial operators from equations (6.1a) and (6.1b) are discretized using the RBF-FD method as it is described in Section 1.4. The RBF chosen is the Gaussian (GA) (see Appendix A), which belongs to the class of infinitely smooth RBFs containing a free shape parameter ε. Since the RBF-FD method only depends on the distances between nodes, the shape and dimension of the domain is not a drawback and it can be easily implemented to solve the current three-dimensional problem. As in [38, 88], we use large stencil sizes and specify the value of the shape parameter such that the matrix condition number is bounded in the range $\kappa_{min} \leq \kappa \leq \kappa_{max}$. This procedure circumvents the intrinsic ill-conditioned of the method and leads to high order approximations as it is shown in Section 1.6.

Therefore, the discretized equations (6.1a) and (6.1b) are given by

$$\frac{\partial \underline{T}}{\partial t} + V(t)\, D_z \underline{T} = \hat{\Delta}\underline{T} + \underline{\omega}, \tag{6.5a}$$

$$\frac{\partial \underline{Y}}{\partial t} + V(t)\, D_z \underline{Y} = \hat{\Delta}\underline{Y} - \underline{\omega}, \tag{6.5b}$$

where D_z and $\hat{\Delta}$ are the RBF-FD differentiation matrices which respectively approximate $\frac{\partial}{\partial z}$ and Δ. $\underline{T}$, $\underline{Y}$ and $\underline{\omega}$ represent functions T, Y and ω evaluated at centers $\{\mathbf{x}_j\}_{j=1}^{N}$.

6.3.2 Time discretization

In order to prevent the flame from leaving the computational domain as time evolves, we follow the method described in [60, 61]. In these works, the authors attach the frame of reference to some point $\mathbf{x}^* = (r^*, \phi^*, z^*)$ that moves with the forefront of the flame with a speed $V(t_k)$ relative to the solid wall by imposing at this point an arbitrary constant temperature in the range $0.2 < T^* < 0.5$ and $\partial T/\partial z|_{\mathbf{x}^*} \neq 0$. Under these constraints, equation (6.5a) at the reference point yields

$$V(t_k)\, D_z \underline{T}^* = \hat{\Delta}\underline{T}^* + \underline{\omega}^*.$$

This equation is used at every time step to compute the velocity of the flame $V(t_k)$. After an initial transient period, if the flame propagates with a constant velocity, the temperature distribution becomes steady in the frame of reference attached to the flame, and the value $V(t_k)$ becomes time independent. This value is the constant flame speed relative to the wall. The criterion for a steady distribution is that

$$\max \frac{\left|\underline{T}^{k+1} - \underline{T}^{k}\right|}{\Delta t} < 10^{-5},$$

where $\underline{T}^k$ and $\underline{T}^{k+1}$ are the values of the temperatures at previous and current time levels, respectively. Since the error committed in determining the velocity with this method is $O(\Delta t)$, equations (6.5a) and (6.5b) are advanced in time using a first order method in time. It results in the following system of equations

$$\underline{T}^{k+1} = \underline{T}^{k} + \Delta t \left[\hat{\Delta}\underline{T}^{k} + \underline{\omega}^{k} - V(t_k)\, D_z \underline{T}^{k}\right], \tag{6.6a}$$

$$\underline{Y}^{k+1} = \underline{Y}^{k} + \Delta t \left[\hat{\Delta}\underline{Y}^{k} - \underline{\omega}^{k} - V(t_k)\, D_z \underline{Y}^{k}\right], \tag{6.6b}$$

where the solution for all interior nodes at time step $k+1$ is computed using the values of the solution at time step k.

Neumann boundary conditions are implemented at every time step in a way similar to that described in [88]. For instance, consider the adiabatic condition

$$\left.\frac{\partial T}{\partial r}\right|_{\mathbf{x}_i} = \left.\left(\frac{\partial T}{\partial x}n_x^i + \frac{\partial T}{\partial y}n_y^i\right)\right|_{\mathbf{x}_i} = 0, \tag{6.7}$$

where $\mathbf{n}^i = (n_x^i, n_y^i, 0)$ is a normal vector to the surface of the duct at one boundary point $\mathbf{x}_i$. Let α_x and α_y be the RBF-FD weights that respectively discretize the first derivatives with respect to x and y on a stencil formed by one boundary point $\mathbf{x}_i$ and $n-1$ interior nodes and let $\{\sigma_i(j)\}_{j=1}^n$ be the set which contains the corresponding node indices. The boundary value $T(\mathbf{x}_i) = \underline{T}_{\sigma_i(1)}$ at time t_{k+1} is given by

$$\underline{T}_{\sigma_i(1)}^{k+1} = -\sum_{j=2}^{n} \frac{\left(n_x^i\alpha_x^j + n_y^i\alpha_y^j\right)}{n_x^i\alpha_x^1 + n_y^i\alpha_y^1}\,\underline{T}_{\sigma_i(j)}^{k+1},$$

which is calculated from the interior values $\{\underline{T}_{\sigma_i(j)}^{k+1}\}_{j=2}^n$ already updated from (6.5a).

6.4 Numerical results

In the following sections, we present the results of solving the mathematical model for different values of the Lewis number Le and the duct radius R using a high-order RBF-FD method. We consider both the two-dimensional and the three-dimensional forms of the model, comparing our results with those presented in [60] where the same model has been solved numerically using the classical second-order finite difference formulas.

6.4.1 2D-results

In this section we present the results obtained with the two-dimensional form of the model where only coordinates z and r are taken into account and the azimuthal coordinate ϕ is not considered. Hence, the laplacian in equations (6.1a) and (6.1b) takes the form $\Delta = \frac{\partial^2}{\partial z^2} + \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r}$, where the variables z and r are the longitudinal and radial coordinates, respectively. Because of the symmetry across the duct centerline, we only consider the region $0 < r < R$

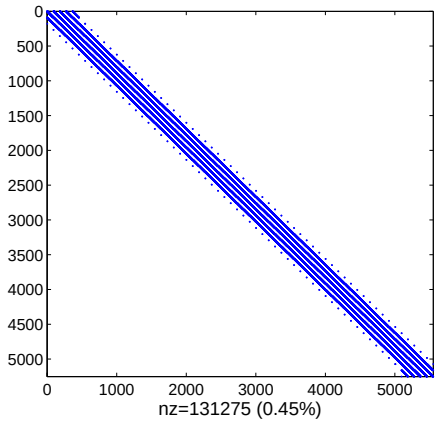


Figure 6.1: Sparsity of the differentiation matrix $\hat{\Delta}$ using a grid of 91×61 nodes and a 25 node stencil.

adding the boundary conditions

$$\left.\frac{\partial Y}{\partial r}\right|_{r=0} = 0 \qquad \text{and} \qquad \left.\frac{\partial T}{\partial r}\right|_{r=0} = 0.$$

In our simulations, the upstream and downstream boundaries are placed at $z_{min} = -10$ and $z_{max} = 20$. The resulting domain is discretized using a grid of 91×61 nodes. The stencil size is $n = 25$, which results in an accuracy corresponding roughly with a fourth-order method. The time step size is $\Delta t = 10^{-3}$ except for the case $Le = 0.5$, where we use $\Delta t = 0.5 \cdot 10^{-3}$ to ensure numerical stability. For all stencils, the RBF-FD weights are computed solving (1.18) with condition numbers in the range $10^{12} \leq \kappa \leq 10^{14}$. In Figure 6.1 we represent the non-zero elements of the differentiation matrix $\hat{\Delta}$. Notice the high sparsity of RBF-FD for which only 0.45 percent of the elements of the matrix are non-zero.

Figure 6.2 shows the solution for the case $Le = 1$ and $R = 15$ with isothermal boundary conditions (compare with Figure 2 in [60]). The flame structure has the so-called mushroom-shape. Notice that the flame extinguishes near the wall and the reaction rate increases smoothly towards the axis of the tube, where the flame structure is almost planar. Figures 6.3 and 6.4 show solutions corresponding to $R = 20$ and 6, with $Le = 0.7$ and isothermal walls (compare with Figure 3 and 4 in [60]). Notice that for $R = 20$ a two cell solution exists with a structure similar to the tulip flame structure: the forefront of the flame is located near the wall and the flame almost extinguishes near the axis. For $R = 6$, the single cell leading edge is located at the

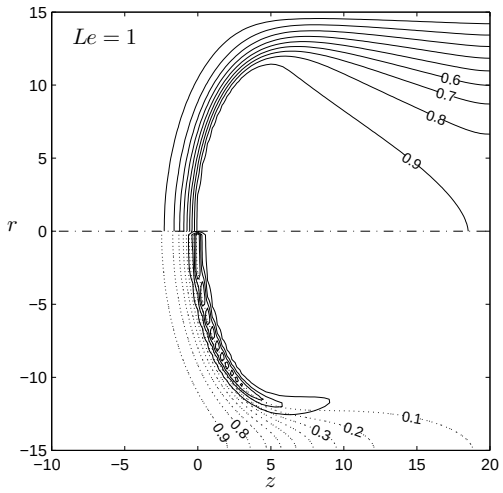


Figure 6.2: Flame structure for $Le = 1$, $R = 15$ and isothermal wall. Upper half: isotherms (solid lines: T at intervals 0.1, $T_{min} = 0.1$ and $T_{max} = 0.9$). Lower half: fuel mass fraction contours (dotted lines: Y at intervals 0.1, $Y_{min} = 0.1$) and reaction rate contours (solid lines: $\omega = 0.1$, 0.5, 1).

axis of the tube, reaching the maximum temperature just behind the flame front. The corresponding time evolution of the non-dimensional flame velocities and numerical tolerances are plotted in Figure 6.5. In both cases, the velocity becomes time independent and the tolerance satisfies the criterion for a steady distribution. Similar behavior is observed in all the simulations presented in this work. Figures 6.6 and 6.7 (compare with Figure 5 and 6 in [60]) show the adiabatic and isothermal wall solutions for large values of the radius in the case $Le = 0.5$. In both cases it is observed a two cell structure with a similar appearance near the axis of the tube. Nevertheless, the isothermal wall solution produces quenching near the wall and, as a result, the structure of the flame is different in that region.

As it is shown, the problems analyzed in this section evolves with the same qualitative features than those presented in [60].

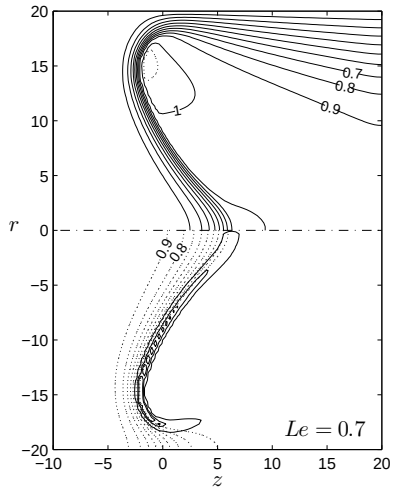


Figure 6.3: Flame structure for $Le = 0.7$, $R = 20$ and isothermal wall. Upper half: isotherms (solid lines: T at intervals 0.1, $T_{min} = 0.1$ and $T_{max} = 1$; dotted lines: $T = 1.02$). Lower half: fuel mass fraction contours (dotted lines: Y at intervals 0.1, $Y_{min} = 0.1$) and reaction rate contours (solid lines: $\omega = 0.1$, 1, 2).

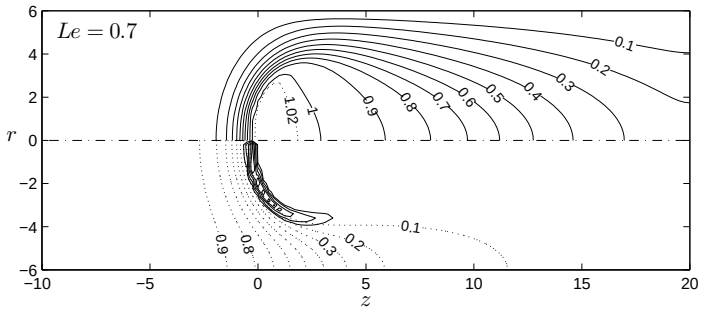


Figure 6.4: Flame structure for $Le = 0.7$, $R = 6$ and isothermal wall. Upper half: isotherms (solid lines: T at intervals 0.1, $T_{min} = 0.1$ and $T_{max} = 1$; dotted lines: $T = 1.02$). Lower half: fuel mass fraction contours (dotted lines: Y at intervals 0.1, $Y_{min} = 0.1$) and reaction rate contours (solid lines: $\omega = 0.5$, 1, 2, 3, 4).

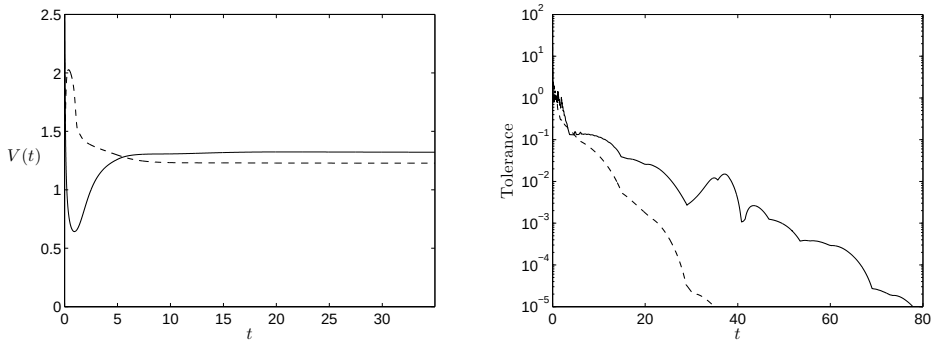


Figure 6.5: Time evolution of the non-dimensional flame velocity (left) and numerical tolerance (right) for $Le = 0.7$ and $R = 20$ (solid line) and $R = 6$ (dashed line).

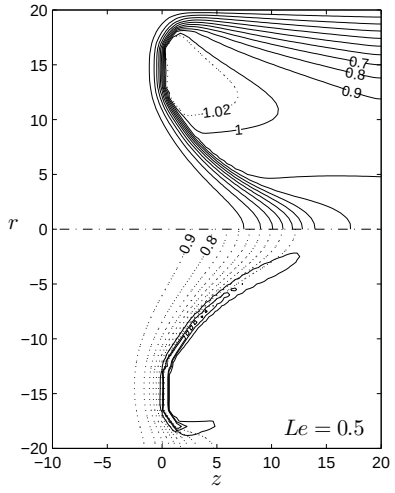


Figure 6.6: Flame structure for $Le = 0.5$, $R = 20$ and isothermal wall. Upper half: isotherms (solid lines: T at intervals, $T_{min} = 0.1$ and $T_{max} = 1$; dotted lines: $T = 1.02$, 1.04 and 1.06). Lower half: fuel mass fraction contours (dotted lines: Y at intervals 0.1, $Y_{min} = 0.1$) and reaction rate contours (solid lines: $\omega = 0.1$, 1, 2).

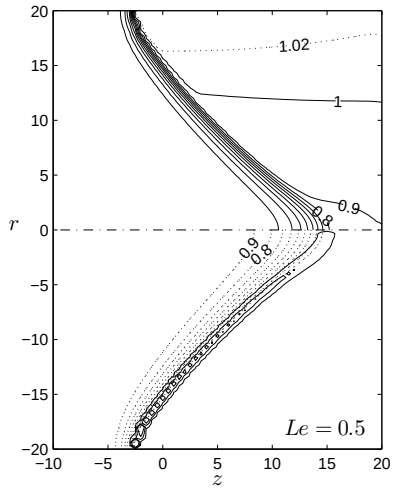


Figure 6.7: Flame structure for $Le = 0.5$, $R = 20$ and adiabatic wall. Upper half: isotherms (solid lines: T at intervals 0.1 and $T_{min} = 0.1$). Lower half: fuel mass fraction contours (dotted lines: Y at intervals 0.1, $Y_{min} = 0.1$) and reaction rate contours (solid lines: $\omega = 0.1$, 1, 2).

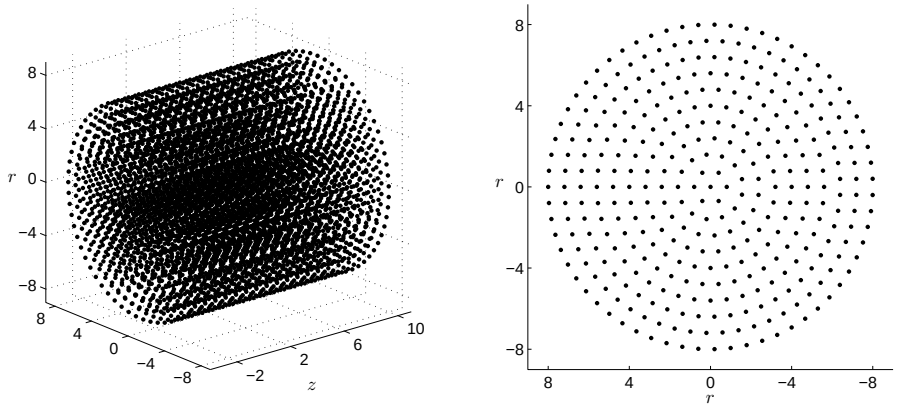


Figure 6.8: Three-dimensional center distribution with $N = 6300$ (left) and transversal section (right).

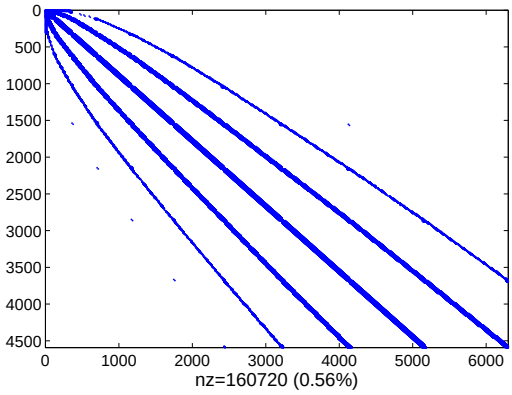


Figure 6.9: Sparsity of the differentiation matrix $\hat{\Delta}$ using the node distribution shown in Figure 6.8 and a 35 node stencil.

6.4.2 3D-results

In this subsection we present the results of solving the three-dimensional model in a duct of radius $R = 8$ extending from $z_{min} = -3$ to $z_{max} = 10$, where the upstream and downstream boundaries are respectively placed. This domain is discretized using 6300 nodes distributed as it is shown in Figure 6.8. The stencil size is $n = 35$, which in three dimensions corresponds roughly with a third-order method. The time step size is $\Delta t = 0.5 \cdot 10^{-3}$ to ensure numerical stability. For all stencils, the RBF-FD weights are computed solving (1.18) with condition numbers in the range $10^{12} \leq \kappa \leq 10^{14}$. Figure 6.9 shows the non-zero elements of the differentiation matrix $\hat{\Delta}$ with these settings. As in the two-dimensional case, the matrix is highly sparse with only 0.56 percent of the elements non-zero.

Depending on the initial conditions, the model can reach different steady state solutions for the same set of parameters [48, 51, 60]. For instance, Figures 6.10 and 6.11 show the steady state flame structures computed for $Le = 0.5$ and $R = 8$ with isothermal boundary conditions using different initial conditions. Figure 6.10 shows the isosurface $T = 0.7$ (left) and the longitudinal section of the flame (right). Both reveal an axisymmetric structure similar to that obtained in the corresponding two-dimensional case shown in Figure 6.6, despite the different radius. Figure 6.11 shows the isosurface $T = 0.9$ (left) and a longitudinal section of the flame (right). In this case, a complicated flame structure with no axial symmetry is obtained. The right side of the figure reveals that the maximum temperature is reached inside the lobules of the flame. Figures 6.12 and 6.13 show respectively the solutions

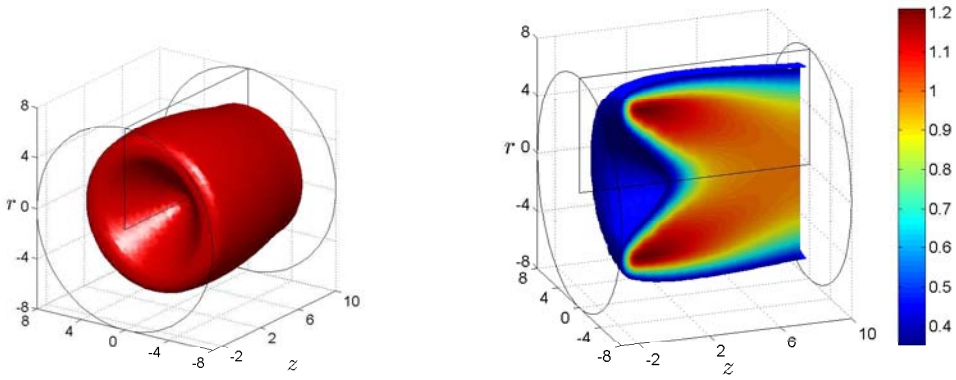


Figure 6.10: Three-dimensional flame structure for $Le = 0.5$, $R = 8$ and isothermal wall. Left: isosurfaces $T = 0.7$. Right: longitudinal section of the flame.

for $Le = 0.5$ and $R = 8$ with adiabatic boundary conditions and $Le = 0.7$ and $R = 8$ with isothermal boundary conditions. Notice that in both cases the flame structure is not axisymmetric.

Comparing these results with those analyzed in [60], it is observed that the RBF-FD solution evolves with the same qualitative features than those computed with FD. The main advantage of RBF-FD over standard FD is that, as the *Mairhuber-Curtis Theorem* [29, 78] points out, in FD it is not possible to compute the weights on arbitrary unstructured stencils in more than 1D, while in RBF methods this problem does not exist [79, 92] (see Chapter 1). As a result, RBF-FD is able to solve problems using an arbitrary node distribution with high accuracy in space by just increasing the number of nodes in the stencil and/or by choosing appropriately the shape parameter.

6.5 Conclusions

In this chapter we analyze the use of RBF-FD methods to numerically solve a model for the propagation of flames in open circular ducts. Since the formulation only depends on the distance between centers and not on their location, the method is essentially the same for any dimension and shape of the domain. Given an arbitrary node distribution, the RBF-FD method is able to calculate the weighting coefficients that approximate the differential operator by just solving the system of equations (1.18) for every stencil, which is guaranteed to be non-singular. We take advantage of this feature

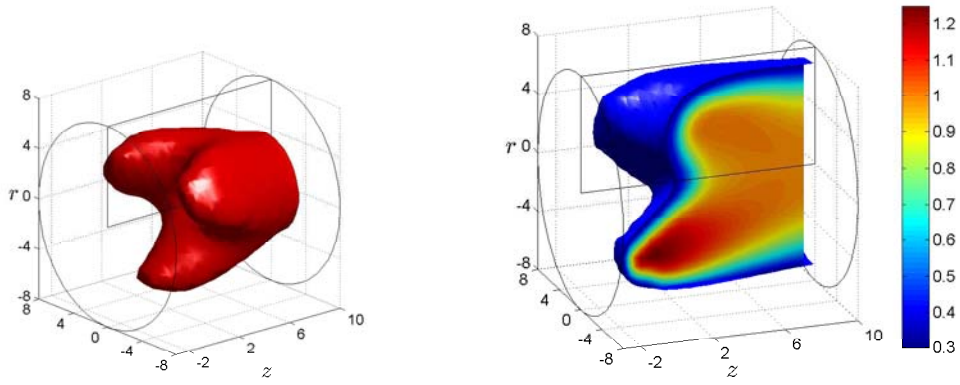


Figure 6.11: Three-dimensional flame structure for $Le = 0.5$, $R = 8$ and isothermal wall. Left: isosurfaces $T = 0.9$. Right: longitudinal section of the flame.

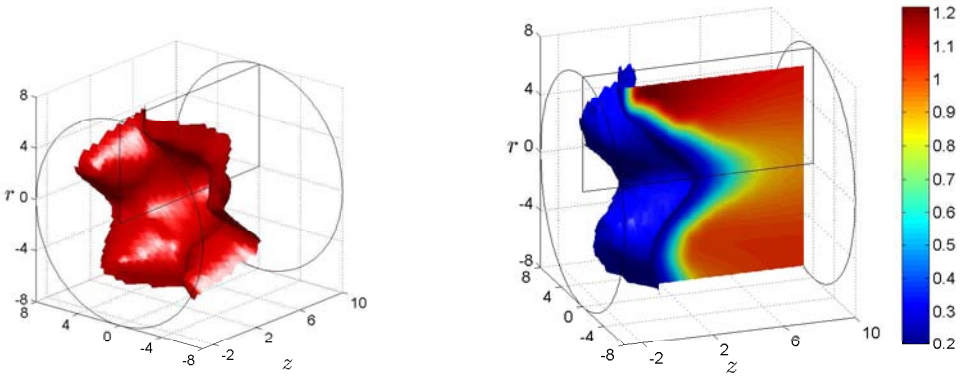


Figure 6.12: Three-dimensional flame structure for $Le = 0.5$, $R = 8$ and adiabatic wall. Left: isosurfaces $T = 0.8$. Right: longitudinal section of the flame.

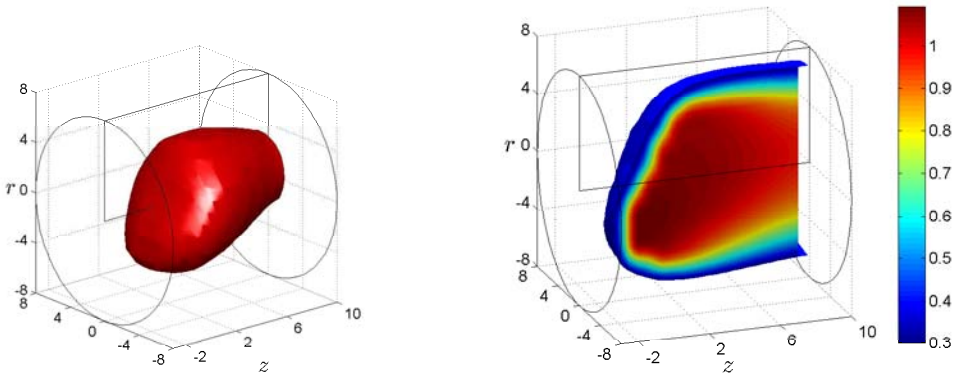


Figure 6.13: Three-dimensional flame structure for $Le = 0.7$, $R = 8$ and isothermal wall. Left: isosurfaces $T = 0.9$. Right: longitudinal section of the flame.

and implement the method for the solution of a three-dimensional problem using a meshless discretization.

As in FD, the order of the approximation depends on the stencil size. However, RBFs contain a free shape parameter which modifies the accuracy of the approximation and the conditioning number of the system of equations. It is more accurate for smaller values of the shape parameter for which the system of equations is ill-conditioned. Hence, accuracy and ill-conditioning cannot be kept simultaneously small. In order to control this trade off, we follow the strategy proposed in [38, 88] to select the shape parameter. It is based on specifying a value of the shape parameter on each stencil so that the matrix condition number is bounded, $\kappa_{min} \leq \kappa \leq \kappa_{max}$. This strategy yields a value of the shape parameter in the region for which the system of equations (1.18) is still well-conditioned and the solution is computed accurately.

Unlike the global RBF method, the resulting differentiation matrices are highly sparse and well-conditioned. In the examples considered, around the 0.5 percent of the elements were non-zero with the corresponding savings in time and memory with respect to the global RBF method. The solutions computed are compared to those obtained in [60], showing the same qualitative features.

From the results obtained, it is concluded that RBF-FD methods are suitable to efficiently solve large scales problems in complexly *nth*-dimensional shaped domains.

Chapter 7

An RBF-FD method for the analysis of an idealized micro-rotary engine

7.1 Introduction

The miniaturization of electromechanical devices and the resulting need for micro-power these systems has motivated the recent development of the field of microscale combustion. The idea is to use the high specific energy of liquid hydrocarbon fuels in combustion-driven microdevices to generate power. A miniature combustion device with a power conversion efficiency of around $2-5\,\%$ would be comparable in its performance to the best available battery [102].

Several microcombustors have been already built and operate [97, 101, 104] and the development of this technology has advanced through the feasibility stage. Nevertheless, there are important issues concerned with heat losses, flame instabilities or flame quenching that need to be addressed to make this technology practical and feasible. Attempts to overcome these difficulties in fabrication has been primarily done by trial and error.

Among the different microcombustors for small-power generation, rotary engine are specially well suited due to its high specific power, low cost and minimum number of moving parts [96]. An example of a micro-rotary engine is represented in Figure 7.1. In order to analyze the behavior of the microcombustor, a numerical study is required. In this chapter we present a simplified mathematical model of an idealized micro-rotary engine whose irregular geometry makes the RBF-FD method suitable for the numerical study.

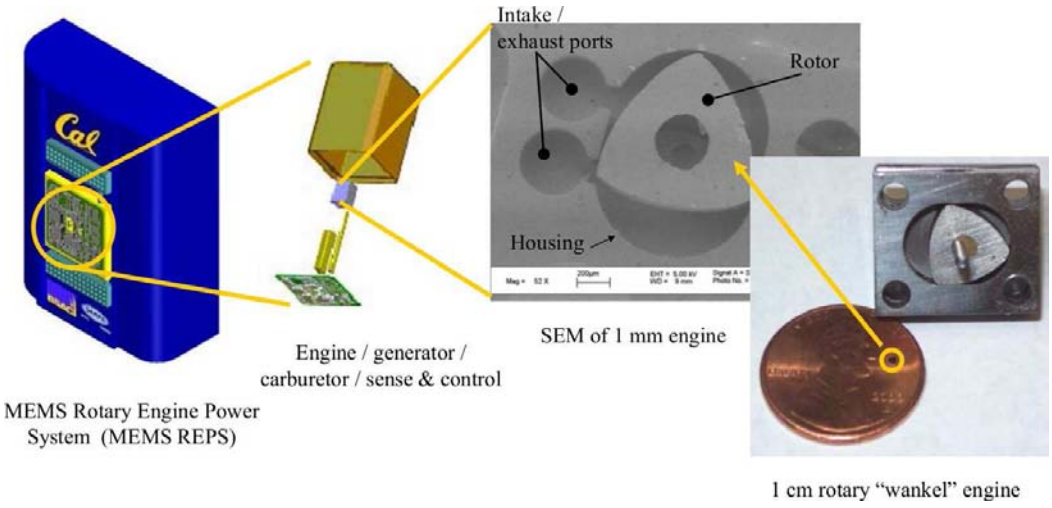


Figure 7.1: Sketch of a micro-rotary engine from the *Micro-Rotary Combustion Lab, University of California, Berkeley.*

7.2 Formulation

In this model, the combustion cavity is approximated by a channel of width h through which a combustible mixture (fuel and oxidizer) flows. The bottom wall moves with a velocity $\pm V$ relative to the other. The upper wall has a notch which modifies the combustible flow and facilitates the attachment of the flame. The velocity profile at the inlet, far away from the notch, is assumed to be a sum of a Poiseuille flow and a Couette flow with a mean velocity U_0. The density and temperature of the fresh mixture are ρ_0 and T_0 and the fuel mass fraction is Y_0. Two different geometries are considered, depending if the notches are inner or outer to the channel. Both are represented in Figure 7.2, where the notches are circumferences of radius R_c and center at (x_c, y_c) which intersects the duct.

When the flow pass through the channel, a recirculation zone appears due to the notch (Figure 7.2). If the mixture is ignited, a steady flame is established in the channel. Its structure and location depend on the flow rate which determine the attachment position, among other parameters. To modelize this problem, a diffusive-thermal model is used, formally assuming that the mixture density ρ, viscosity μ, thermal diffusivity D_T, heat capacity c_p, and fuel molecular diffusivity D_F are all constants. Consequently, the flow field is not affected by the combustion process and can be determined a-priori by solving the steady Navier-Stokes equation in the given channel configuration. The dimensionless variables are defined using h and h^2/D_T as units of

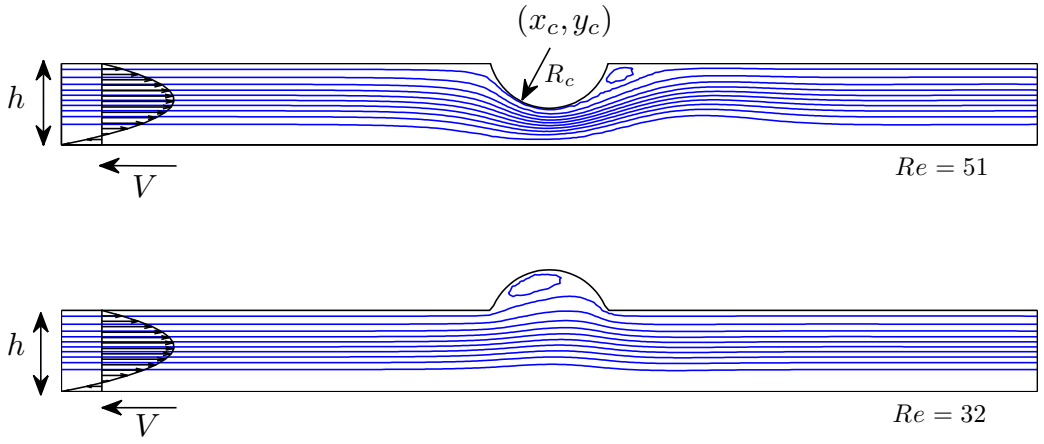


Figure 7.2: Sketches of channel configurations for inner notch (top) and outer notch (bottom). The flow field is illustrated by selected streamlines. The notches are modelized by circumference of radius R_c centered at (x_c, y_c).

length and time, and U_0 as a unit of velocity. The coordinates x and y are dimensionless in units of h. The mass fraction is normalized with respect to Y_0 and a non-dimensional temperature is introduced as $\theta = (T - T_0)/(T_1 - T_0)$, where $T_1 = T_0 + QY_0/c_p$ is the corresponding adiabatic flame temperature. The laminar flame speed S_L and the thermal flame thickness $\delta_T = D_T/S_L$ are additional characteristic scales which introduce two parameters: the mass flow rate parameter $m = \rho U_0/\rho S_L$ and the square of the channel width to the laminar flame thickness $d = (h/\delta_T)^2$. The governing equations in dimensionless form are given by

$$\nabla \cdot \mathbf{u} = 0 \tag{7.1a}$$

$$m\sqrt{d}\,(\mathbf{u}\cdot\nabla)\,\mathbf{u} = -\nabla p + Pr\,\nabla^2\mathbf{u} \tag{7.1b}$$

$$\frac{\partial\theta}{\partial t} + m\sqrt{d}\,(\mathbf{u}\cdot\nabla)\,\theta = \nabla^2\theta + d\cdot\omega\,(\theta, Y) \tag{7.1c}$$

$$\frac{\partial Y}{\partial t} + m\sqrt{d}\,(\mathbf{u}\cdot\nabla)\,Y = \frac{1}{Le}\nabla^2 Y - d\cdot\omega\,(\theta, Y) \tag{7.1d}$$

where $\mathbf{u} = (u, v)$ is the velocity vector with u, v the axial and transverse components, p is the pressure, Y is the fuel mass fraction and

$$\omega\,(\theta, Y) = \frac{Ze^2}{2\,Le\;{u_p}^2}\,Y \exp\left[\frac{Ze\,(\theta-1)}{1+\gamma\,(\theta-1)}\right]. \tag{7.2}$$

The non-dimensional parameters appearing in these equations are the Zeldovich number, $Ze = E(T_a - T_0)/RT_a^2$, the Lewis number, $Le = D_T/D_F$,

the heat release parameter, $\gamma = (T_a - T_0)/T_a$, and the Prandtl number, $Pr = \mu/\rho D_T$, where E is the activation energy of the chemical reaction and R the gas constant. The Reynolds number, based on the mean flow velocity U_0 is expressed as $Re = m\sqrt{d}/Pr$. The factor $u_p = S_L/U_L$ arises in the non-dimensional reaction rate (7.2) where U_L is the asymptotic value of the velocity of the planar flame obtained for large activation energies ($Ze \gg 1$) and S_L is the planar burning velocity. The non-dimensional boundary conditions along the duct walls are,

$$\begin{aligned} y = 0: \qquad & u = \bar{V},\ v = 0, \qquad && \frac{\partial \theta}{\partial \vec{n}} = \tilde{b}\sqrt{d}\ \theta, \quad \frac{\partial Y}{\partial \vec{n}} = 0, \\ y = y_s(x): \qquad & u = v = 0, \qquad && \frac{\partial \theta}{\partial \vec{n}} = \tilde{b}\sqrt{d}\ \theta, \quad \frac{\partial Y}{\partial \vec{n}} = 0. \end{aligned}$$

where $y_s(x)$ denotes the surface of the irregular upper wall, $\tilde{b} = b\delta_T$ is the non-dimensional heat loss coefficient and $\tilde{V} = V/U_0$ is the non-dimensional velocity of the bottom wall. At the inlet, far upstream as $x \to -\infty$,

$$\begin{cases} u(y) = 6y(1-y) + 3\tilde{V}(y-1)(y-1/3),\ v = 0 \\ Y = 1,\ \theta = 0 \\ \dfrac{\partial p}{\partial x} = 0 \end{cases}$$

and at the outlet, far downstream as $x \to +\infty$,

$$Pr\,\frac{\partial u}{\partial x} = p, \qquad \frac{\partial v}{\partial x} = \frac{\partial Y}{\partial x} = \frac{\partial \theta}{\partial x} = 0.$$

The pressure conditions are taken from [86, 87].

7.3 Numerical implementation

The numerical solution of the steady Navier-Stokes equations (7.1a) and (7.1b) involves some difficulties, like the lack of an independent equation for the pressure and the non-existence of a dominant variable in the continuity equation. To circumvent these difficulties, we make use of the fractional step algorithm originally suggested by Chorin [16]. In this algorithm, the pressure is uncoupled from the momentum equations and construct so that the flow field satisfies the continuity equation. In this way, equation (7.1b) is splitted

using the two-fractional step formulation as

$$\frac{\mathbf{u}^{n+1/2}-\mathbf{u}^{n}}{\Delta t} = -m\sqrt{d}\,(\mathbf{u}^{n}\cdot\nabla)\,\mathbf{u}^{n}+P_r\,\nabla^2\mathbf{u}^{n} \tag{7.3a}$$

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n+1/2}}{\Delta t} = -\nabla p^{n+1} \tag{7.3b}$$

where the superscripts denotes the time levels. In this approach, time plays the role of an iterative parameter and only the state obtained in the last iteration has physical meaning. The steady state is assumed to be achieved whenever it is satisfied the tolerance condition $\max|\mathbf{u}^{n+1}-\mathbf{u}^{n}|/\Delta t < \text{tol}$.

Equation (7.3a) is discretized in space using RBF-FD formulas and advanced in time using a semi-implicit AB2CN scheme [21] (second-order Adam-Bashford scheme for the advection term and Crank-Nicolson scheme for the diffusive term). It is well-known that the implicit treatment of diffusion term eliminates the viscous stability constraint which can be quite severe in numerical computations of viscous flow, while the explicit treatment of the advection term makes only necessary to solve a linear boundary problem at each time step [6]. Therefore, the intermediate velocity $\mathbf{u}^{n+1/2}$ is calculated from

$$\frac{\mathbf{u}^{n+1/2}-\mathbf{u}^{n}}{\Delta t} = -\frac{m\sqrt{d}}{2}\left[3\mathbf{H}^{n}-\mathbf{H}^{n-1}\right]+\frac{P_r}{2}\left(L\,\mathbf{u}^{n+1/2}+L\,\mathbf{u}^{n}\right), \tag{7.4}$$

where $\mathbf{H}^{n}$ is given at the *nth* time step by

$$\mathbf{H}^{n} = m\sqrt{d}\,(u^{n}\circ D_x+v^{n}\circ D_y)\,\mathbf{u}^{n}. \tag{7.5}$$

In the equation above, $\circ$ denotes element-wise matrix multiplication and D_x, D_y and L are the respective RBF-FD differentiation matrices that approximates ∂_x, ∂_y and ∇^2. As a pre-processing step, $\left(I\pm\Delta t\frac{Pr}{2}L\right)$ are calculated initially and stored.

The complete velocity $\mathbf{u}^{n+1}$ is corrected by including the pressure field given by (7.3b). At each time step, the solution is expected to satisfy the continuity equation (7.1a). Applying the divergence to equation (7.3b), it is satisfied $\nabla\cdot\mathbf{u}^{n+1}=0$ and pressure is given by the Poisson equation

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t}\nabla\cdot\mathbf{u}^{n+1/2}, \tag{7.6}$$

with boundary conditions along the walls

$$\frac{\partial p^{n+1}}{\partial\vec{n}} = \frac{1}{\Delta t}\left(\mathbf{u}^{n+1/2}-\mathbf{u}^{n}\right)\cdot\vec{n}. \tag{7.7}$$

Finally, the velocity $\mathbf{u}^{n+1}$ is updated substituting the solution of pressure equation in (7.3b).

To validate the numerical implementation, we have considered the classical problem of a lid-driven flow in a square cavity solved using RBF-FD methods in [14, 21, 87]. The streamlines and vorticity are shown in Figure 7.3 for (a) $Re = 100$ and (b) $Re = 400$. As it is appreciated, the solution computed evolves with the same qualitative features than those from the cited works.

Once the flow field has reached the steady state, the stationary flow field $\mathbf{u}^s = (u^s, v^s)$ is substituted in equations (7.1c - 7.1d) and both are advanced in time using an semi-implicit AB3CN scheme,

$$\frac{\theta^{n+1}-\theta^n}{\Delta t} = \left[\tfrac{23}{12}F^n - \tfrac{16}{12}F^{n-1} + \tfrac{5}{12}F^{n-2}\right] + \left(L\,\theta^{n+1} + L\,\theta^n\right) \tag{7.8a}$$

$$\frac{Y^{n+1}-Y^n}{\Delta t} = \left[\tfrac{23}{12}G^n - \tfrac{16}{12}G^{n-1} + \tfrac{5}{12}G^{n-2}\right] + \tfrac{1}{Le}\left(L\,Y^{n+1} + L\,Y^n\right) \tag{7.8b}$$

where F^n and G^n are given at the *nth* time step by

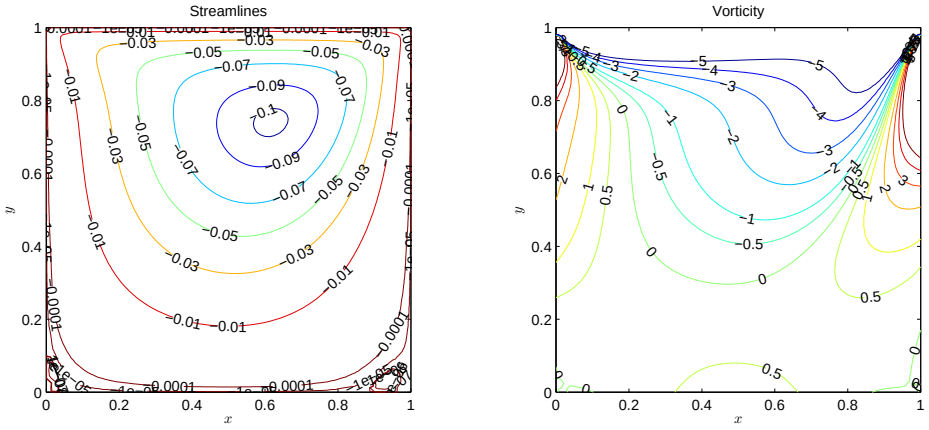
$$\begin{aligned} F^n &= m\sqrt{d}\,\left(u^s \circ D_x + v^s \circ D_y\right)\theta^n + d\cdot\omega\left(\theta^n, Y^n\right), \\ G^n &= m\sqrt{d}\,\left(u^s \circ D_x + v^s \circ D_y\right)Y^n - d\cdot\omega\left(\theta^n, Y^n\right). \end{aligned} \tag{7.9}$$

Here, $\circ$ denotes element-wise matrix multiplication and the RBF-FD differentiation matrices D_x, D_y and L are equal to those from equations (7.4) and (7.5). A RK4 method is used to initialize the algorithm. As in the combustion problem from Chapter 6, time plays the role of the iterative parameter and only the state obtained in the last iteration has physical meaning. The criterion for a steady distribution is that $\max\left|\theta^{n+1} - \theta^n\right|/\Delta t < \text{tol}$.

7.4 Numerical Results

Motivated from the microcombustor applications, the steady flow field is determined solving the steady Navier-Stokes equations for $0 \leq m < 10$ and keeping fix the channel width h to $d = (h/\delta_T)^2 = 20$. The notches inside the conduct are circumferences of radius $R_c = 0.75$ centered at $(0, 1.2)$ in the outer notch case, and $(0, 0.75)$ in the inner notch case. The Prandtl and Lewis numbers have been assigned the fixed values $Pr = 0.7$ and $Le = 1$. The Zeldovich number and the heat release ratio are $Ze = 7$ and $\gamma = 0.7$. For these parameter values, the value for the laminar flame speed is $u_p = 0.8920$. In the following, we consider adiabatic walls for which the heat loss coefficient is $\tilde{b} = 0$.

(a)



(b)

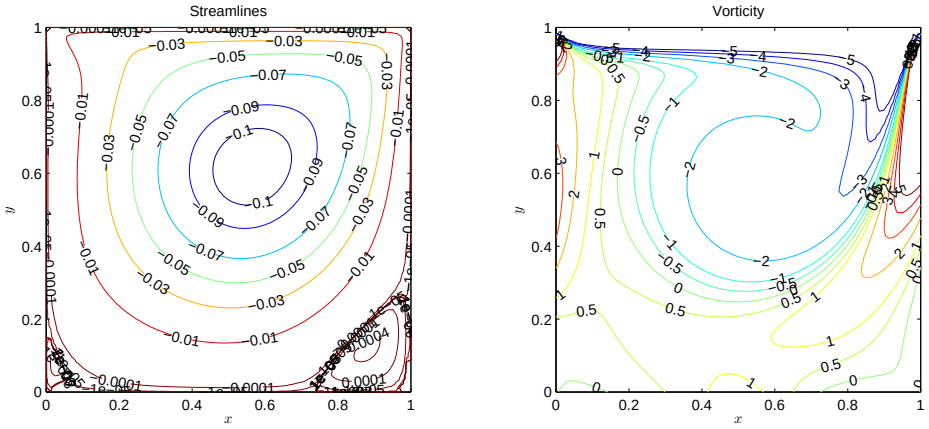


Figure 7.3: Lid-driven cavity problem solved using a semi-implicit RBF-FD formulation for (a) $Re = 100$ and (b) $Re = 400$.

In Figure 7.4 it is represented the results obtained for the inner notch, with a mass flow parameter $m = 2$ and a wall velocity $\tilde{V} = -0.5$. The different plots are: (a) pressure, (b) longitudinal velocity, (c) transversal velocity, (d) vorticity and (e) streamlines of the flow and reaction rate (7.2). Notice how the notch modifies the velocity field creating a low pressure region just behind it. This region coincides with the position where the flame is attached.

In Figure 7.5 it is shown the results for the outer notch case computed with the same set of parameters. In this case, the attachment position also coincides with the low pressure region created by the notch.

The performance of the microcombustor will depend on the attachment position and the length of the flame. Hence, it is important to analize the influence of the mass flow parameter m and wall velocity $\tilde{V}$ on it. For instance, in Figure 7.6 it is represented the reaction rate (7.2) and streamlines for different mass flows and wall velocities. Notice how different are the lengths and shapes of the flames in each case. Although the work is still preliminary, the final objective is to develop a deeper analysis which allows to understand the performance of the microcombustor as a function of all the different parameters.

7.5 Conclusions

In this Chapter we have addressed the numerical study of an idealized micro-rotary engine. Concretely, we have implemented a simplified model in which the microcombustor cavity is a conduct with an irregular geometry through which the combustible flows. In this approach, the mixture density ρ, viscosity μ, thermal diffusivity D_T, heat capacity c_p, and fuel molecular diffusivity D_F are assumed to be constants. Consequently, the flow field is not affected by the combustion process and can be determined a-priori by solving the steady Navier-Stokes equation in the given channel configuration. The combustion is then approximated by a diffusive-thermal model.

Due to the irregular geometry of the problem, the RBF-FD method is used to approximate the spatial operators. Time discretization of Navier-Stokes equations are carried out using an AB2CN scheme, while the diffusive-thermal model is solved using an AB3CN scheme. The implicit treatment eliminates the viscous stability constraint which can be quite severe in numerical computations of viscous flow. Although preliminary, the results point out the success of the RBF-FD method handling problems with irregular geometries. The next step of this work is focused on understanding the performance of the microcombustor as a function of all the different parameters.

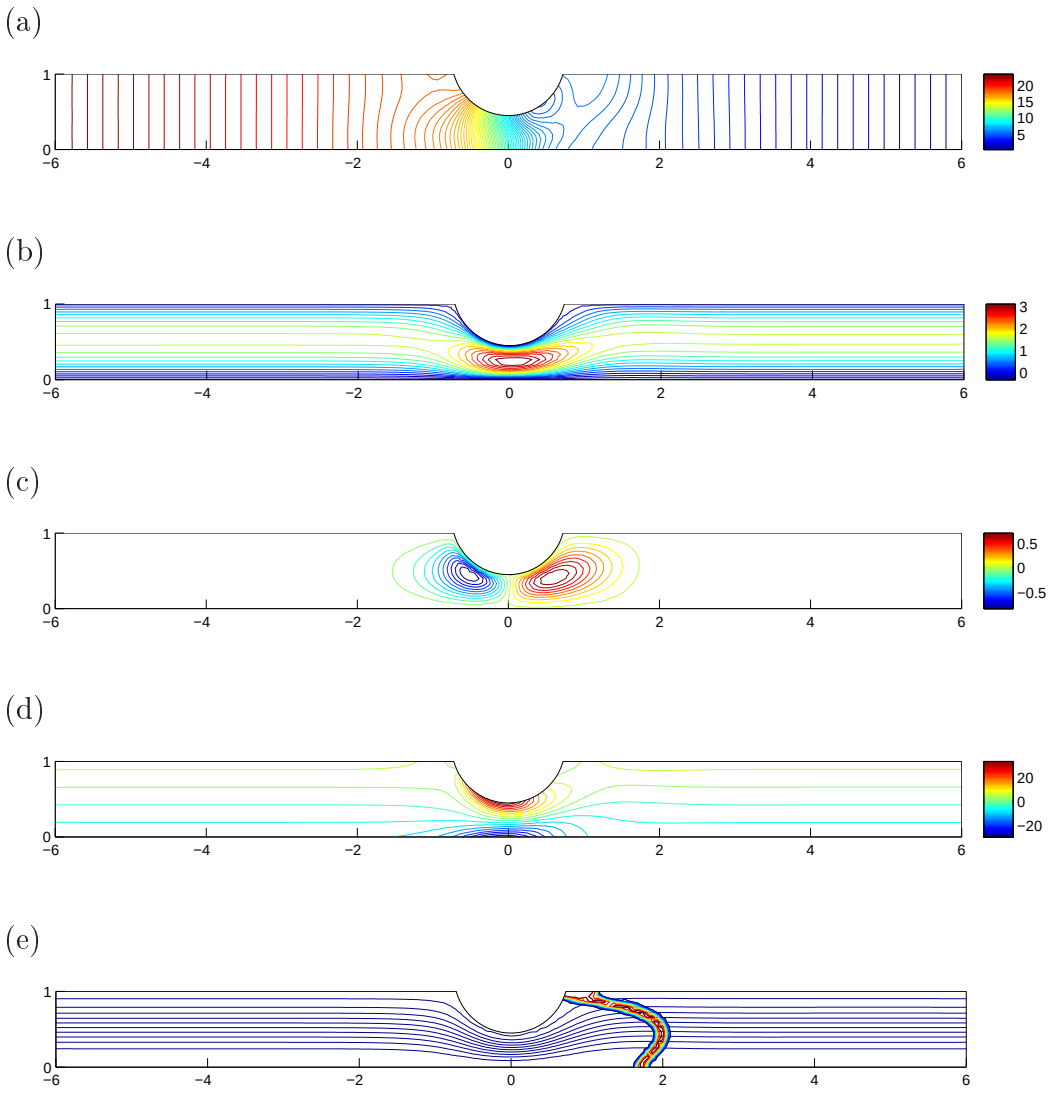


Figure 7.4: From top to bottom: (a) pressure, (b) u-velocity, (c) v-velocity, (d) vorticity, (e) streamlines and reaction rate (7.2), calculated for a mass flow rate parameter $m = 2$ and wall velocity $\tilde{V} = -0.5$.

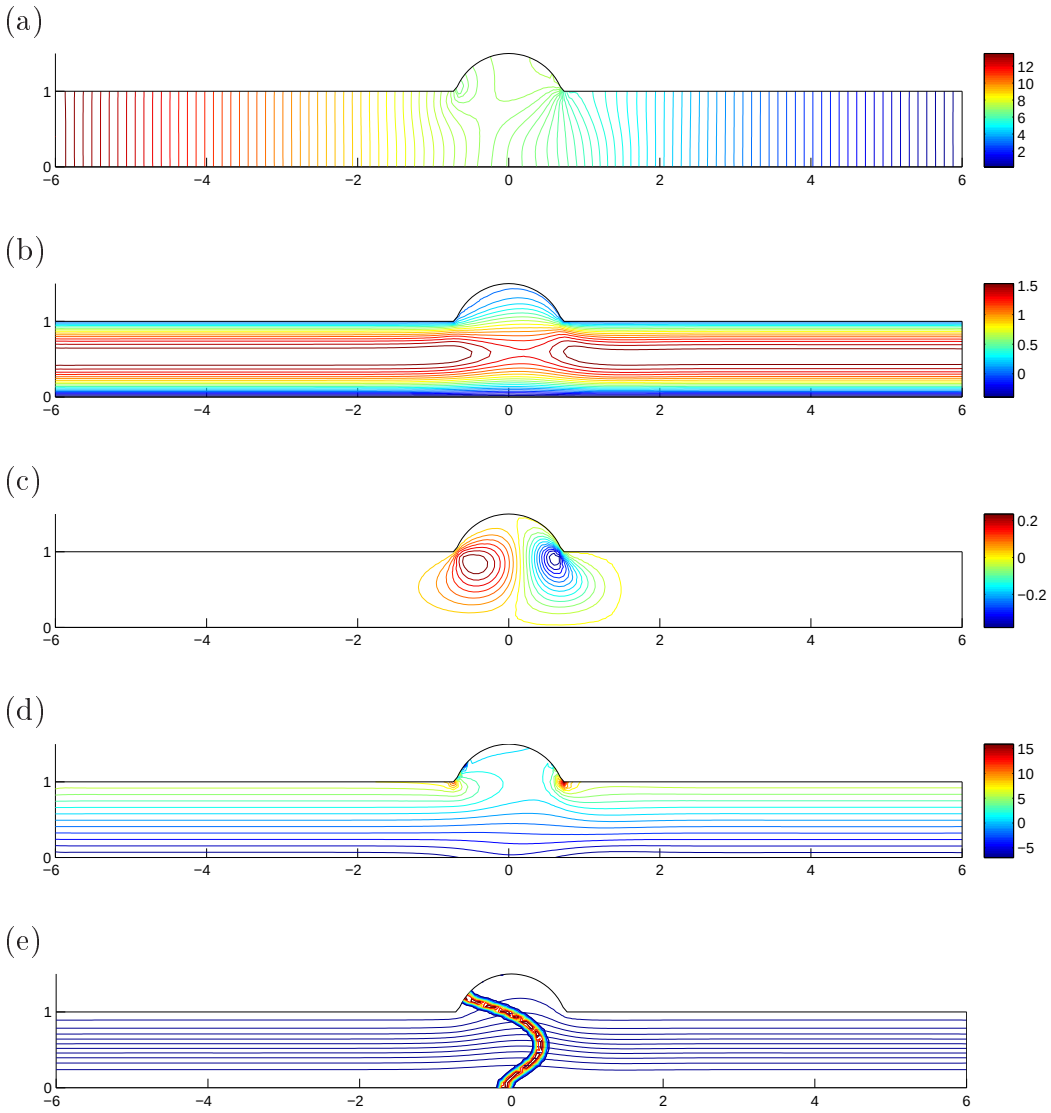


Figure 7.5: From top to bottom: (a) pressure, (b) u-velocity, (c) v-velocity, (d) vorticity, (e) streamlines and reaction rate (7.2), calculated for a flow with $m = 2$ and velocity wall $\tilde{V} = -0.5$.

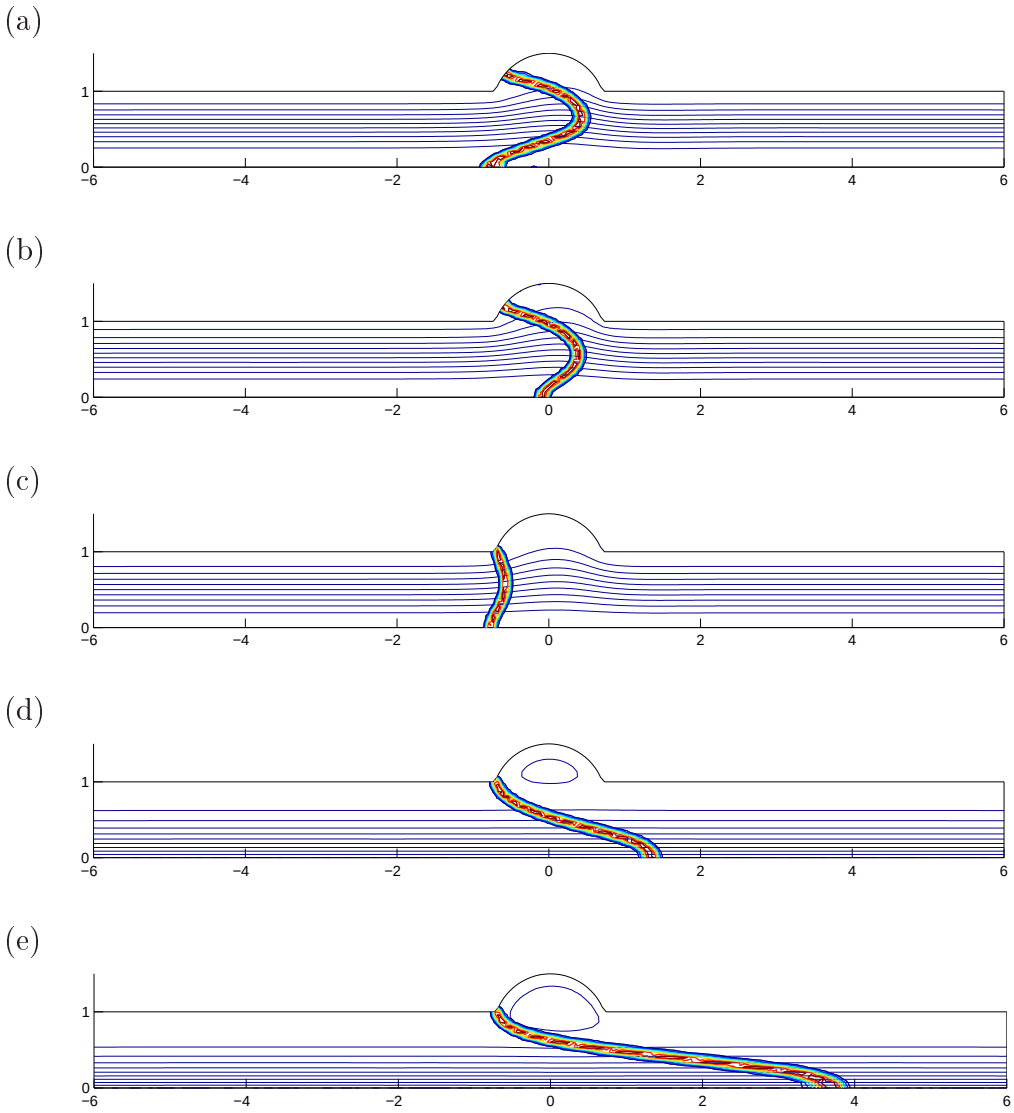


Figure 7.6: From top to bottom: (a) $m = 2.7$ and $\tilde{V} = -0.8$, (b) $m = 2$ and $\tilde{V} = -0.5$, (c) $m = 1.1$ and $\tilde{V} = 0$, (d) $m = 2.45$ and $\tilde{V} = 2.5$, (e) $m = 4.8$ and $\tilde{V} = 3$.

Conclusions

In this thesis the RBF-FD method for the solution of PDEs is considered. In the first part we have analyzed the convergence properties of the method obtaining novel analytical formulas for the local truncation error as a function of the shape parameter, inter-nodal distance and stencil size (Chapter 2 and Appendix A). This result proves the existence of a range of values of the shape parameter for which RBF-FD methods are more accurate than FD. Indeed, it usually exists an optimal shape parameter for which the terms of the local truncation error cancel out and the approximation is exact. To leading order, such a value is independent of the inter-nodal distance and only relies on the function and its derivatives.

These results allow the development of novel algorithms for the selection of the shape parameter in the solution of PDEs. Two different strategies are proposed: a *node-independent shape parameter*, which minimizes the norm of the global error (Chapter 3), and a *node-dependent shape parameter*, which minimizes the local truncation error at each node of the domain (Chapter 4). Applications of the present methods have been studied from academic problems (Chapter 3 and 4) to the solution of classical elastostatic problems (Chapter 5), for which it is shown that the accuracy can significantly increased one or two orders of magnitude with respect to finite differences by efficiently tuning the values of shape parameters.

The applicability of the method in the resolution of practical problems is also considered in the second part of this thesis. Its main feature, i.e. the ability of handling irregular domains using highly sparse differentiation matrices while approximating the differential operators to high order, makes the method specially well suited for this purpose. In this way, a three-dimensional problem for the propagation of a premixed laminar flame through a duct has been solved (Chapter 6). The good performance of the method inspires us to implement an RBF-FD method for the numerical study of an idealized Wankel microcombustor (Chapter 7), for which the geometry is more complex. The combustible flow field and the combustion process are respectively modeled through the steady Navier-Stokes equation and the combus-

tion model from Chapter 6.

There are several lines of investigation still open for further work. For instance, the algorithms proposed for the selection of the optimal shape parameter are not all that suitable for unstructured nodes. The procedure relies on computing an intermediate FD solution which allows the estimation of the optimal value. However, for two dimensions or higher, selecting the stencil nodes to approximate high order derivatives might be problematic and time consuming. Instead, the global RBF method on a coarse node distribution could be used as an alternative.

On the other hand, there are no exact analytical formulas available to estimate the optimal shape parameter for large stencil sizes. Increasing the size of the stencil expands the system of equations which determines the RBF-FD weights, making unattainable the analytical solution. These problems should be addressed in the future to make the algorithms more practical.

The equivalence between global and local RBF methods shown in Chapter 1 allows us to expand the convergence properties studied in this thesis also to the global case. If exact error formulas are finally derived for large stencil sizes, novel algorithms for the selection of the optimal shape parameter will also be developed for the global method.

In the field of applications, we will continue with the analysis of the numerical model for the micro-rotary engine. The main objective is to determine the parameter ranges in which the engine performs and to analyze which type of notch (inner or outer) works best. From the preliminary results, we find the RBF-FD method specially suitable to deal with practical problems for which the complexity of the geometry domain and/or the order of accuracy are constraints of the problem.

Part I

Appendix

Appendix A

Gaussian RBF-FD formulas

A.1 Outline

In this appendix, we extend the analysis presented in Chapter 2 and consider the convergence properties of RBF-FD and RBF-HFD formulas using Gaussians as RBFs (see Table 1.1). In this way, we derive analytical expressions of the weights for first and second order derivatives in 1D, and for the Laplacian in 2D, using equispaced nodes in all cases. Only first and second order derivatives formulas in 1D are derived for the RBF-HFD method. The weights are functions of the inter-nodal distance h and the shape parameter ε. Contrary to what happened with multiquadrics in Chapter 2, where the weights were written as Taylor series expansions in powers of h, for Gaussians it is often possible to write them as short analytical formulas. These coefficients are then used to derive analytical expressions for the leading term of the local truncation error in the limit $\varepsilon h \ll 1$.

The validity of the analytical formulas are illustrated through numerical examples, where we use

$$u(\mathbf{x}) \,=\, \sin\left(\|\mathbf{x}\|^2\right)$$

as test function, where $\|\mathbf{x}\|$ is the euclidean norm. Equations (1.18) and (1.24) are used to compute the coefficients needed to approximate $\mathcal{L}u$ at $x_0 = 0.4$ and $\mathbf{x}_0 = (0.4, 0.4)$ in 1D and 2D, respectively. For each formula we compute the absolute value of the error as a function of the shape parameter ε and the inter-nodal distance h, and compare it with the leading term of the local truncation error derived in the limit $\varepsilon h \ll 1$.

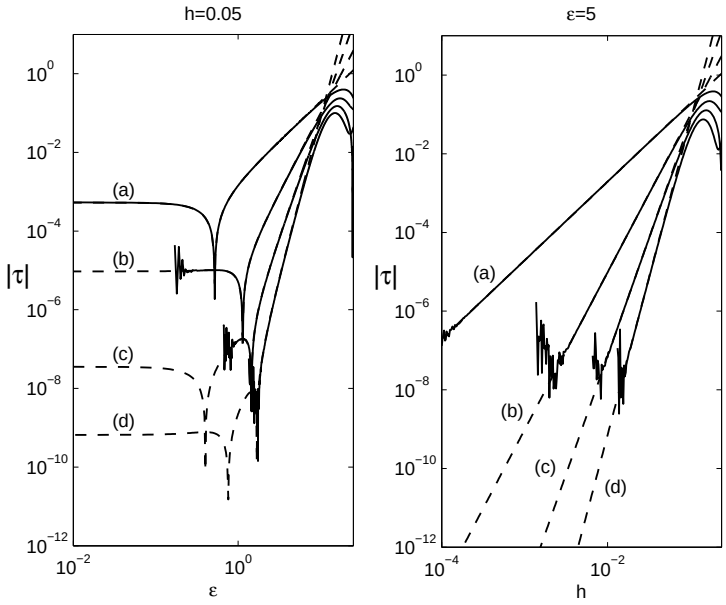


Figure A.1: Local truncation error (τ_n) for the RBF-FD first derivative as function of ε (left side) and h (right side) using structured stencils with (a) $n = 3$, (b) $n = 5$, (c) $n = 7$, and (d) $n = 9$ nodes. Solid lines: local truncation error computed solving numerically (1.18). Dashed lines: leading order formulas of the errors given in Table A.1.

A.2 RBF-FD formulas

A.2.1 First derivative

Tables A.1 and A.2 show the weights and the corresponding local truncation errors for RBF-FD formulas to approximate the first derivative in 1D, respectively. Exact expressions are given for 3, 5 and 7 equispaced nodes. For 9 equispaced nodes the exact formulas are too long and therefore we only include their series expansions in the limit $\varepsilon h \ll 1$. The results for 3 and 5 nodes are in agreement with those previously derived in Appendix A of reference [7].

Figure A.1 shows the corresponding error (solid line) for $n = 3$, 5, 7 and 9 when the weights are computed by solving numerically the linear system (1.18). This error is compared with the approximate error given by the formulas in Table A.2 (dashed line). Notice that the agreement is excellent up to the point where the linear system to numerically compute the weights (1.18) becomes ill-conditioned and round-off errors deteriorate the accuracy

Table A.1: RBF-FD weights for the first derivative.

Three nodes	
α_0	0
$\alpha_{\pm1}$	$\pm\frac{1}{2}\varepsilon^2 h\left(\operatorname{csch}\left(\varepsilon^2h^2\right)+\operatorname{sech}\left(\varepsilon^2h^2\right)\right)$
Five nodes	
α_0	0
$\alpha_{\pm1}$	$\pm\varepsilon^2 h\left(1+e^{2\varepsilon^2h^2}\right)\operatorname{csch}\left(3\varepsilon^2h^2\right)$
$\alpha_{\pm2}$	$\mp\frac{\varepsilon^2 h e^{4\varepsilon^2h^2}}{\sinh(2\varepsilon^2h^2)+\sinh(4\varepsilon^2h^2)+\sinh(6\varepsilon^2h^2)}$
Seven nodes	
α_0	0
$\alpha_{\pm1}$	$\pm\varepsilon^2 h\left(e^{3\varepsilon^2h^2}+2\cosh\left(\varepsilon^2h^2\right)\right)\operatorname{csch}\left(4\varepsilon^2h^2\right)$
$\alpha_{\pm2}$	$\mp\frac{\varepsilon^2 h e^{2\varepsilon^2h^2}\left(1+e^{2\varepsilon^2h^2}+e^{4\varepsilon^2h^2}\right)}{\sinh(2\varepsilon^2h^2)+\sinh(4\varepsilon^2h^2)+\sinh(6\varepsilon^2h^2)+\sinh(8\varepsilon^2h^2)}$
α_{+3}	$\frac{\varepsilon^2 h e^{9\varepsilon^2h^2}}{\sinh(2\varepsilon^2h^2)+2\sinh(4\varepsilon^2h^2)+2\sinh(6\varepsilon^2h^2)+2\sinh(8\varepsilon^2h^2)+\sinh(10\varepsilon^2h^2)+\sinh(12\varepsilon^2h^2)}$
α_{-3}	$-\frac{\varepsilon^2 h e^{9\varepsilon^2h^2}\operatorname{csch}\left(6\varepsilon^2h^2\right)}{2(1+2\cosh(2\varepsilon^2h^2)+\cosh(4\varepsilon^2h^2)+\cosh(6\varepsilon^2h^2))}$
Nine nodes	
α_0	0
$\alpha_{\pm1}$	$\pm\frac{4}{5h}\pm\frac{4\varepsilon^2h}{5}\mp\frac{14\varepsilon^4h^3}{15}\mp\frac{6\varepsilon^6h^5}{5}\pm\frac{191\varepsilon^8h^7}{90}+O\left(\varepsilon^{10}h^9\right)$
$\alpha_{\pm2}$	$\mp\frac{1}{5h}\mp\frac{4\varepsilon^2h}{5}\mp\frac{4\varepsilon^4h^3}{15}\pm\frac{16\varepsilon^6h^5}{5}\pm\frac{104\varepsilon^8h^7}{45}+O\left(\varepsilon^{10}h^9\right)$
$\alpha_{\pm3}$	$\pm\frac{4}{105h}\pm\frac{12\varepsilon^2h}{35}\pm\frac{34\varepsilon^4h^3}{35}\mp\frac{18\varepsilon^6h^5}{35}\mp\frac{531\varepsilon^8h^7}{70}+O\left(\varepsilon^{10}h^9\right)$
$\alpha_{\pm4}$	$\mp\frac{1}{280h}\mp\frac{2\varepsilon^2h}{35}\mp\frac{38\varepsilon^4h^3}{105}\mp\frac{32\varepsilon^6h^5}{35}\pm\frac{316\varepsilon^8h^7}{315}+O\left(\varepsilon^{10}h^9\right)$

Table A.2: RBF-FD local truncation errors for the first derivative.

	Local truncation error
τ_3	$\frac{h^2}{6}\left(u'''(x_0)+6\varepsilon^2 u'(x_0)\right)+O\left(h^4 P_2(\varepsilon^2)\right)$
τ_5	$-\frac{h^4}{30}\left(u^{(V)}(x_0)+20\varepsilon^2 u'''(x_0)+60\varepsilon^4 u'(x_0)\right)+O\left(h^6 P_3(\varepsilon^2)\right)$
τ_7	$\frac{h^6}{140}\left(u^{(VII)}(x_0)+42\varepsilon^2 u^{(V)}(x_0)+420\varepsilon^4 u'''(x_0)+840\varepsilon^6 u'(x_0)\right)+O\left(h^8 P_4(\varepsilon^2)\right)$
τ_9	$-\frac{h^8}{630}\left(u^{(IX)}(x_0)+72\varepsilon^2 u^{(VII)}(x_0)+1512\varepsilon^4 u^{(V)}(x_0)+10080\varepsilon^6 u'''(x_0)+15120\varepsilon^8 u'(x_0)\right)+O\left(h^{10}P_5(\varepsilon^2)\right)$

of the numerical solution. It should be emphasized that, it is not necessary to numerically solve (1.18) in order to get the weights. Instead, the analytic formulas given in Table A.1 can be directly used.

The left part of Figure A.1 shows the absolute value of the error as a function of the shape parameter for $h = 0.05$. The accuracy increases with decreasing ε. For small ε (flat RBFs) it is well known that RBF-FD formulas approach standard finite difference formulas [22]. This fact can be clearly observed in the figure, which shows how the error approaches the standard finite difference error when $\varepsilon \to 0$. Notice also that there is a range of values of the shape parameter for which RBF-FD formulas are more accurate than standard finite differences. In particular, there is an optimal value, ε^*, for which the local truncation error is zero to leading order. Such value can be accurately estimated from the formulas in Table A.2. The right part of Figure A.1 shows the absolute value of the error as a function of the inter-nodal distance h for $\varepsilon = 5$. Notice that the error behaves as $O(h^{n-1})$ in agreement with the formulas in Table A.2.

A.2.2 Second derivative

Tables A.3 and A.4 show the weights for RBF-FD formulas to approximate the second derivative in 1D using the standard formulation which is not exact for constants (1.18) and the one which is exact for constants (1.25), respectively. Note that in the first case, exact expressions are given for 3, 5 and 7 equispaced nodes. For 9 equispaced nodes only their series expansions in the limit $\varepsilon h \ll 1$ are included. For RBF-FD formulas exact for constants (1.25) exact expressions are only given for 3 equispaced nodes.

Tables A.5 and A.6 show the corresponding local truncation errors. Note that the term of the error independent of the shape parameter ε coincides in

both formulations and it is equivalent to the FD error. Notice also that the errors in the formulation which is non exact for constants (Table A.5), there are some extra terms proportional to $\varepsilon^{n+1}u(x_0)$. This is expected since it is not exact for constants. Thus, for values of ε of order unity or larger, both formulations may differ significantly.

Figure A.2 shows the numerical error (solid line) in the approximation of the second derivative for $n = 3, 5, 7, 9$ using the formulation (1.18) which is not exact for constants (top) and the formulation (1.25) which is exact for constants (bottom). The numerical results are compared with the approximate error given by the formulas in Table A.5 and A.6 (dashed line). The left part of Figure A.2 shows the absolute value of the error as a function of the shape parameter for $h = 0.05$, and the right part shows the absolute value of the error as a function of the inter-nodal distance h for $\varepsilon = 5$. In the first case, the accuracy increases with decreasing ε and approaches standard finite differences for small ε. In both cases the results coincide until the system of equations (1.18) or (1.25) becomes ill-conditioned. Notice that there is an optimal shape parameter ε^* which makes the error zero to leading order.

Table A.3: RBF-FD weights for the second derivative: non exact for constants

Three nodes	
α_0	$-2\left(\varepsilon^2+\varepsilon^4h^2\mathrm{csch}^2\left(\varepsilon^2h^2\right)\right)$
$\alpha_{\pm1}$	$\varepsilon^4h^2\left(1+\coth\left(\varepsilon^2h^2\right)\right)\mathrm{csch}\left(\varepsilon^2h^2\right)$
Five nodes	
α_0	$\frac{1}{2}\left(\varepsilon^4h^2\left(\mathrm{sech}^2\left(\varepsilon^2h^2\right)-5\mathrm{csch}^2\left(\varepsilon^2h^2\right)\right)-4\varepsilon^2\right)$
$\alpha_{\pm1}$	$\frac{4\varepsilon^4h^2\cosh\left(\varepsilon^2h^2\right)\coth\left(\varepsilon^2h^2\right)\left(\coth\left(\varepsilon^2h^2\right)+1\right)}{2\cosh(2\varepsilon^2h^2)+1}$
$\alpha_{\pm2}$	$-\frac{\varepsilon^4h^2e^{4\varepsilon^2h^2}\mathrm{csch}^2\left(2\varepsilon^2h^2\right)}{2\cosh(2\varepsilon^2h^2)+1}$
Seven nodes	
α_0	$\frac{1}{18}\varepsilon^2\left(\frac{32\varepsilon^2h^2\left(\cosh\left(2\varepsilon^2h^2\right)+2\right)}{\left(2\cosh(2\varepsilon^2h^2)+1\right)^2}-49\varepsilon^2h^2\mathrm{csch}^2\left(\varepsilon^2h^2\right)+9\varepsilon^2h^2\mathrm{sech}^2\left(\varepsilon^2h^2\right)-36\right)$
$\alpha_{\pm1}$	$\frac{1}{2}\varepsilon^4h^2\left(\coth\left(\varepsilon^2h^2\right)+1\right)\mathrm{csch}\left(\varepsilon^2h^2\right)\left(\mathrm{sech}\left(2\varepsilon^2h^2\right)+2\right)$
$\alpha_{\pm2}$	$-\frac{\varepsilon^4h^2e^{2\varepsilon^2h^2}\left(e^{2\varepsilon^2h^2}+e^{4\varepsilon^2h^2}+1\right)\mathrm{csch}^2\left(2\varepsilon^2h^2\right)}{2\cosh(2\varepsilon^2h^2)+2\cosh(4\varepsilon^2h^2)+1}$
$\alpha_{\pm3}$	$\frac{\varepsilon^4h^2e^{9\varepsilon^2h^2}\mathrm{csch}^2\left(3\varepsilon^2h^2\right)}{2(2\cosh(2\varepsilon^2h^2)+\cosh(4\varepsilon^2h^2)+\cosh(6\varepsilon^2h^2)+1)}$
Nine nodes	
α_0	$-\frac{205}{72h^2}-2\varepsilon^2+\frac{8h^2\varepsilon^4}{3}-4h^6\varepsilon^8+O\left(h^{10}\varepsilon^{12}\right)$
$\alpha_{\pm1}$	$\frac{8}{5h^2}+\frac{8\varepsilon^2}{5}-\frac{4h^2\varepsilon^4}{3}-\frac{28h^4\varepsilon^6}{15}+\frac{269h^6\varepsilon^8}{75}+\frac{191h^8\varepsilon^{10}}{45}+O\left(h^{10}\varepsilon^{12}\right)$
$\alpha_{\pm2}$	$-\frac{1}{5h^2}-\frac{4\varepsilon^2}{5}-\frac{8h^2\varepsilon^4}{15}+\frac{32h^4\varepsilon^6}{15}+\frac{152h^6\varepsilon^8}{75}-\frac{352h^8\varepsilon^{10}}{45}+O\left(h^{10}\varepsilon^{12}\right)$
$\alpha_{\pm2}$	$\frac{8}{315h^2}+\frac{8\varepsilon^2}{35}+\frac{76h^2\varepsilon^4}{105}+\frac{12h^4\varepsilon^6}{35}-\frac{79h^6\mathrm{ep}^8}{25}-\frac{153h^8\varepsilon^{10}}{35}+O\left(h^{10}\varepsilon^{12}\right)$
$\alpha_{\pm4}$	$-\frac{1}{560h^2}-\frac{\varepsilon^2}{35}-\frac{4h^2\varepsilon^4}{21}-\frac{64h^4\varepsilon^6}{105}-\frac{34h^6\varepsilon^8}{75}+\frac{992h^8\varepsilon^{10}}{315}+O\left(h^{10}\varepsilon^{12}\right)$

Table A.4: RBF-FD weights for the second derivative: exact for constants

Three nodes	
α_0	$-\frac{4\varepsilon^2 e^{3\varepsilon^2 h^2}\left(2\varepsilon^2 h^2+e^{\varepsilon^2 h^2}-1\right)}{-4e^{3\varepsilon^2 h^2}+3e^{4\varepsilon^2 h^2}+1}$
$\alpha_{\pm 1}$	$\frac{2\varepsilon^2 e^{3\varepsilon^2 h^2}\left(2\varepsilon^2 h^2+e^{\varepsilon^2 h^2}-1\right)}{-4e^{3\varepsilon^2 h^2}+3e^{4\varepsilon^2 h^2}+1}$
Five nodes	
α_0	$-\frac{5}{2h^2}-\frac{28\varepsilon^2}{15}+\frac{83h^2\varepsilon^4}{90}+O\left(h^4\varepsilon^6\right)$
$\alpha_{\pm 1}$	$\frac{4}{3h^2}+\frac{56\varepsilon^2}{45}-\frac{13h^2\varepsilon^4}{135}+O\left(h^4\varepsilon^6\right)$
$\alpha_{\pm 2}$	$-\frac{1}{12h^2}-\frac{14\varepsilon^2}{45}-\frac{197h^2\varepsilon^4}{540}+O\left(h^4\varepsilon^6\right)$
Seven nodes	
α_0	$-\frac{49}{18h^2}-\frac{27\varepsilon^2}{14}+\frac{237h^2\varepsilon^4}{140}+\frac{199h^4\varepsilon^6}{300}+O\left(h^6\varepsilon^8\right)$
$\alpha_{\pm 1}$	$\frac{3}{2h^2}+\frac{81\varepsilon^2}{56}-\frac{333h^2\varepsilon^4}{560}-\frac{533h^4\varepsilon^6}{400}+O\left(h^6\varepsilon^8\right)$
$\alpha_{\pm 2}$	$-\frac{3}{20h^2}-\frac{81\varepsilon^2}{140}-\frac{801h^2\varepsilon^4}{1400}+\frac{127h^4\varepsilon^6}{200}+O\left(h^6\varepsilon^8\right)$
$\alpha_{\pm 3}$	$\frac{1}{90h^2}+\frac{27\varepsilon^2}{280}+\frac{897h^2\varepsilon^4}{2800}+\frac{439h^4\varepsilon^6}{1200}+O\left(h^6\varepsilon^8\right)$
Nine nodes	
α_0	$-\frac{205}{72h^2}-\frac{88\varepsilon^2}{45}+\frac{254h^2\varepsilon^4}{105}+\frac{358h^4\varepsilon^6}{525}-\frac{173561h^6\varepsilon^8}{33075}+O\left(h^8\varepsilon^{10}\right)$
$\alpha_{\pm 1}$	$\frac{8}{5h^2}+\frac{352\varepsilon^2}{225}-\frac{124h^2\varepsilon^4}{105}-\frac{1832h^4\varepsilon^6}{875}+\frac{569729h^6\varepsilon^8}{165375}+O\left(h^8\varepsilon^{10}\right)$
$\alpha_{\pm 2}$	$-\frac{1}{5h^2}-\frac{176\varepsilon^2}{225}-\frac{284h^2\varepsilon^4}{525}+\frac{1716h^4\varepsilon^6}{875}+\frac{375542h^6\varepsilon^8}{165375}+O\left(h^8\varepsilon^{10}\right)$
$\alpha_{\pm 2}$	$\frac{8}{315h^2}+\frac{352\varepsilon^2}{1575}+\frac{52h^2\varepsilon^4}{75}+\frac{6344h^4\varepsilon^6}{18375}-\frac{457057h^6\varepsilon^8}{165375}+O\left(h^8\varepsilon^{10}\right)$
$\alpha_{\pm 4}$	$-\frac{1}{560h^2}-\frac{44\varepsilon^2}{1575}-\frac{19h^2\varepsilon^4}{105}-\frac{3391h^4\varepsilon^6}{6125}-\frac{108623h^6\varepsilon^8}{330750}+O\left(h^8\varepsilon^{10}\right)$

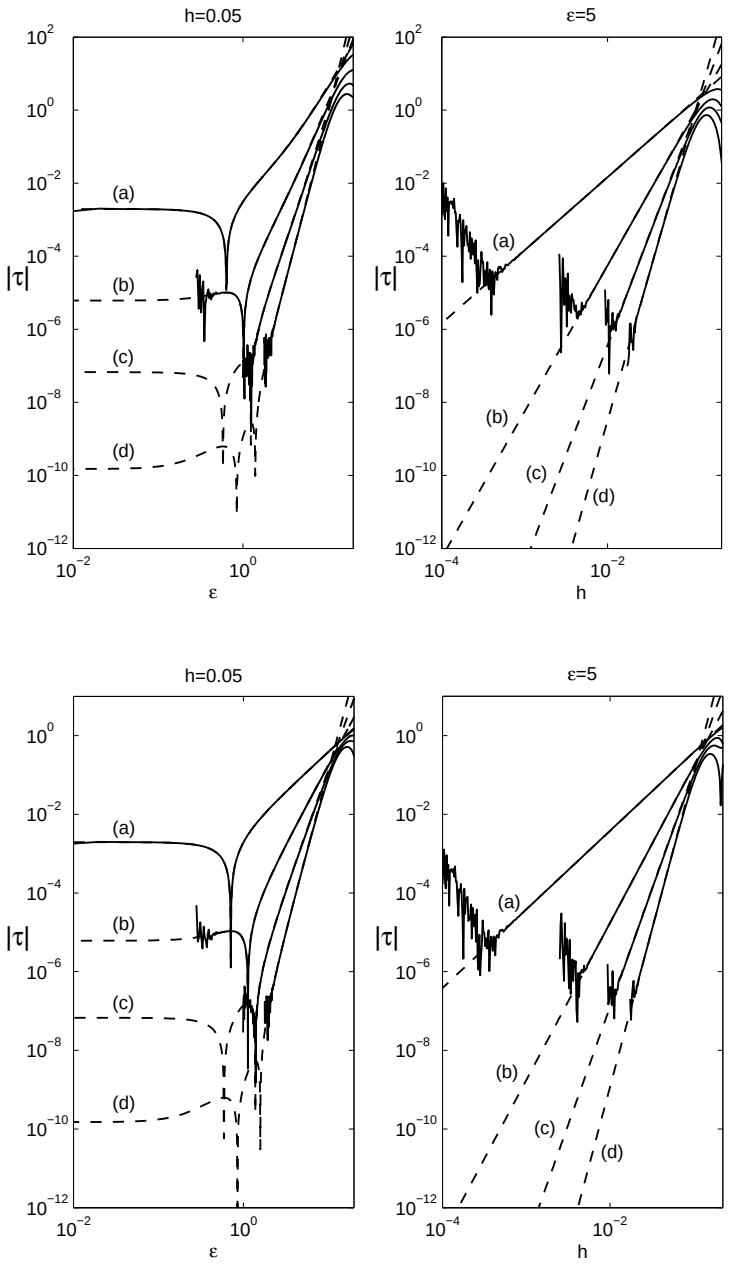


Figure A.2: Top: same as Figure A.1 but for the RBF-FD second derivative (non exact for constants). Bottom: same as Figure A.1 but for the RBF-FD second derivative (exact for constants).

Table A.5: RBF-FD local truncation errors for the second derivative: non exact for constants

	Local truncation error
τ_3	$\frac{h^2}{12}\left(u^{(IV)}(x_0)+12\varepsilon^2 u''(x_0)+12\varepsilon^4 u(x_0)\right)+O\left(h^4 P_3(\varepsilon^2)\right)$
τ_5	$-\frac{h^4}{90}\left(u^{(VI)}(x_0)+30\varepsilon^2 u^{(IV)}(x_0)+180\varepsilon^4 u''(x_0)+120\varepsilon^6 u(x_0)\right)+O\left(h^6 P_4(\varepsilon^2)\right)$
τ_7	$\frac{h^6}{560}\left(u^{(VIII)}(x_0)+56\varepsilon^2 u^{(VI)}(x_0)+840\varepsilon^4 u^{(IV)}(x_0)+3360\varepsilon^6 u''(x_0)+1680\varepsilon^8 u(x_0)\right)+O\left(h^8 P_5(\varepsilon^2)\right)$
τ_9	$-\frac{h^8}{3150}\left(u^{(X)}(x_0)+90\varepsilon^2 u^{(VIII)}(x_0)+2520\varepsilon^4 u^{(VI)}(x_0)+25200\varepsilon^6 u^{(IV)}(x_0)+75600\varepsilon^8 u''(x_0)+30240\varepsilon^{10} u(x_0)\right)+O\left(h^{10} P_6(\varepsilon^2)\right)$

Table A.6: RBF-FD local truncation errors for the second derivative: exact for constants

	Local truncation error
τ_3	$\frac{h^2}{12}\left(u^{(IV)}(x_0)+10\varepsilon^2 u''(x_0)\right)+O\left(h^4 P_2(\varepsilon^2)\right)$
τ_5	$-\frac{h^4}{90}\left(u^{(VI)}(x_0)+28\varepsilon^2 u^{(IV)}(x_0)+140\varepsilon^4 u''(x_0)\right)+O\left(h^6 P_3(\varepsilon^2)\right)$
τ_7	$\frac{h^6}{560}\left(u^{(VIII)}(x_0)+54\varepsilon^2 u^{(VI)}(x_0)+756\varepsilon^4 u^{(IV)}(x_0)+2520\varepsilon^6 u''(x_0)\right)+O\left(h^8 P_4(\varepsilon^2)\right)$
τ_9	$-\frac{h^8}{3150}\left(u^{(X)}(x_0)+88\varepsilon^2 u^{(VIII)}(x_0)+2376\varepsilon^4 u^{(VI)}(x_0)+22176\varepsilon^6 u^{(IV)}(x_0)+55440\varepsilon^8 u''(x_0)\right)+O\left(h^{10} P_5(\varepsilon^2)\right)$

A.2.3 Laplacian

Tables A.7 and A.8 show the weights and the corresponding local truncation errors for the RBF-FD laplacian formulas with 5 and 9 equispaced nodes (non exact and exact for constants, respectively). Notice that in the non exact case (Table A.7), the expressions for the weights and the local error are equivalent in both cases.

Figure A.3 show the local truncation error obtained using the corresponding analytical weights of Tables A.7 and A.8 for $n = 5$ (solid line) and compare it with the local truncation errors obtained numerically with multiquadrics (dashed line). As it is shown in the figure, the rates of convergence are equivalent in both cases. This is due to the fact that the local truncation errors are polynomials of the same degree n in the shape parameter ε (see Section 2.4). As $\varepsilon \to 0$, the local truncation error becomes equivalent since they both approach standard finite differences.

Regarding accuracy, there are not advantages on using either multiquadrics or Gaussians, in general. However, for a specific function there might be significant differences associated to the fact that the location and/or the existence of the optimal shape parameter will change from using either one or the other. In this particular example, the error using multiquadrics is slightly smaller than with Gaussians. Note that the location of the optimal shape parameter is different.

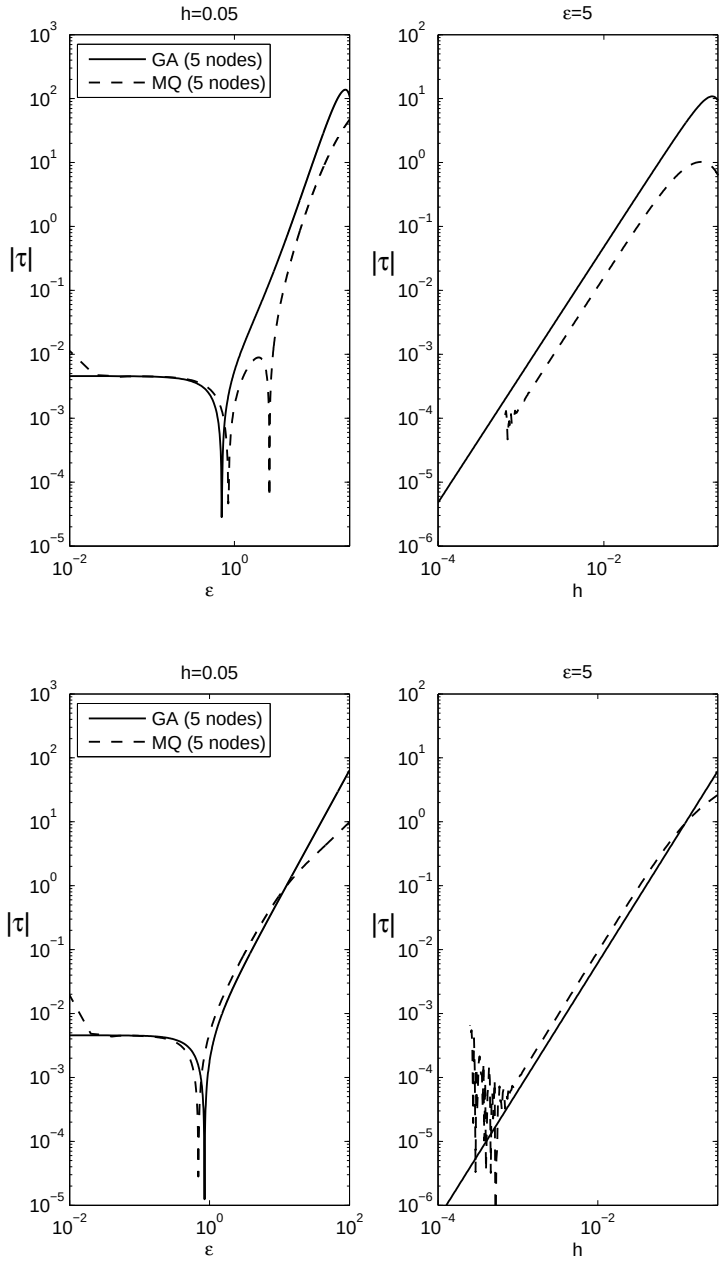


Figure A.3: Top: Local truncation error (τ_5) for the RBF-FD laplacian (non exact for constants) as function of ε (left side) and h (right side) using a structured stencil. Solid line: Gaussians. Dashed line: multiquadrics. Bottom: same as top of the figure but for the RBF-FD laplacian (exact for constants)

Table A.7: RBF-FD laplacian: non exact for constants. Notice that $\tau_9 = \tau_5$.

Five nodes	
α_0	$-4\left(\varepsilon^2 + \varepsilon^4 h^2 \mathrm{csch}^2\left(\varepsilon^2 h^2\right)\right)$
$\alpha_{1,2,3,4}$	$\varepsilon^4 h^2 \left(1 + \coth\left(\varepsilon^2 h^2\right)\right) \mathrm{csch}\left(\varepsilon^2 h^2\right)$
τ_5	$\frac{h^2}{12}\left(u^{(4,0)}(\mathbf{x}_0) + u^{(0,4)}(\mathbf{x}_0)\right) + \varepsilon^2 h^2\left(u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0)\right) + 2\varepsilon^4 h^2 u(\mathbf{x}_0) + O\left(h^4 P_3(\varepsilon^2)\right)$
Nine nodes	
α_0	$-4\left(\varepsilon^2 + \varepsilon^4 h^2 \mathrm{csch}^2\left(\varepsilon^2 h^2\right)\right)$
$\alpha_{1,2,3,4}$	$\varepsilon^4 h^2 \left(1 + \coth\left(\varepsilon^2 h^2\right)\right) \mathrm{csch}\left(\varepsilon^2 h^2\right)$
$\alpha_{5,6,7,8}$	0
τ_9	$\frac{h^2}{12}\left(u^{(4,0)}(\mathbf{x}_0) + u^{(0,4)}(\mathbf{x}_0)\right) + \varepsilon^2 h^2\left(u^{(2,0)}(\mathbf{x}_0) + u^{(0,2)}(\mathbf{x}_0)\right) + 2\varepsilon^4 h^2 u(\mathbf{x}_0) + O\left(h^4 P_3(\varepsilon^2)\right)$

Table A.8: RBF-FD laplacian: exact for constants

Five nodes	
α_0	$-\frac{16\varepsilon^2 e^{3\varepsilon^2 h^2}\left(\varepsilon^2 h^2+e^{\varepsilon^2 h^2}-1\right)}{2e^{2\varepsilon^2 h^2}-8e^{3\varepsilon^2 h^2}+5e^{4\varepsilon^2 h^2}+1}$
$\alpha_{1,2,3,4}$	$\frac{4\varepsilon^2 e^{3\varepsilon^2 h^2}\left(\varepsilon^2 h^2+e^{\varepsilon^2 h^2}-1\right)}{2e^{2\varepsilon^2 h^2}-8e^{3\varepsilon^2 h^2}+5e^{4\varepsilon^2 h^2}+1}$
τ_5	$\frac{h^2}{12}\left(u^{(4,0)}(\mathbf{x}_0)+u^{(0,4)}(\mathbf{x}_0)\right)+\frac{3}{4}\varepsilon^2 h^2\left(u^{(2,0)}(\mathbf{x}_0)+u^{(0,2)}(\mathbf{x}_0)\right)+O\left(h^4 P_2(\varepsilon^2)\right)$
Nine nodes	
α_0	$\frac{16\varepsilon^2 e^{3\varepsilon^2 h^2}\left(\varepsilon^2 h^2+e^{\varepsilon^2 h^2}\left(3\varepsilon^2 h^2+e^{\varepsilon^2 h^2}\left(2\varepsilon^2 h^2+e^{\varepsilon^2 h^2}\left(-2\varepsilon^2 h^2+e^{\varepsilon^2 h^2}\left(-7\varepsilon^2 h^2-e^{\varepsilon^2 h^2}\left(\varepsilon^2 h^2+2e^{\varepsilon^2 h^2}+1\right)+4\right)+3\right)-1\right)-2\right)-1\right)}{\left(e^{\varepsilon^2 h^2}-1\right)^3\left(3e^{\varepsilon^2 h^2}+5e^{2\varepsilon^2 h^2}+3e^{3\varepsilon^2 h^2}+1\right)^2}$
$\alpha_{1,2,3,4}$	$\frac{4\varepsilon^2 e^{3\varepsilon^2 h^2}\left(-\varepsilon^2 h^2+e^{5\varepsilon^2 h^2}+e^{\mathrm{ep}^2 h^2}\left(1-2\varepsilon^2 h^2\right)+2e^{3\varepsilon^2 h^2}\left(\varepsilon^2 h^2-1\right)+e^{4\varepsilon^2 h^2}\left(5\varepsilon^2 h^2-1\right)+1\right)}{\left(e^{\varepsilon^2 h^2}-1\right)^3\left(e^{\varepsilon^2 h^2}+1\right)\left(2e^{\varepsilon^2 h^2}+3e^{2\varepsilon^2 h^2}+1\right)^2}$
$\alpha_{5,6,7,8}$	$\frac{4\varepsilon^2 e^{6\varepsilon^2 h^2}\left(e^{2\varepsilon^2 h^2}\left(-4\varepsilon^2 h^2+2\sinh\left(\varepsilon^2 h^2\right)+1\right)-1\right)}{\left(e^{\varepsilon^2 h^2}-1\right)^3\left(3e^{\varepsilon^2 h^2}+5e^{2\varepsilon^2 h^2}+3e^{3\varepsilon^2 h^2}+1\right)^2}$
τ_9	$\frac{h^2}{36}\left(3u^{(4,0)}(\mathbf{x}_0)-2u^{(2,2)}(\mathbf{x}_0)+3u^{(0,4)}(\mathbf{x}_0)\right)+\frac{2}{3}\varepsilon^2 h^2\left(u^{(0,2)}(\mathbf{x}_0)+u^{(2,0)}(\mathbf{x}_0)\right)+O\left(h^4 P_2(\varepsilon^2)\right)$

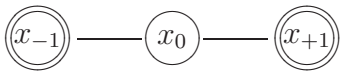
Table A.9: RBF-HFD first derivative

Three nodes	
α_0	0
$\alpha_{\pm 1}$	$\pm\frac{2\varepsilon^2 h e^{3\varepsilon^2h^2}\left(-4\varepsilon^2h^2+e^{4\varepsilon^2h^2}-1\right)}{-8\varepsilon^2h^2e^{4\varepsilon^2h^2}+e^{8\varepsilon^2h^2}-1}$
$\omega_{\pm 1}$	$\frac{e^{\varepsilon^2h^2}\left(2\varepsilon^2h^2\cosh\left(2\varepsilon^2h^2\right)-\sinh\left(2\varepsilon^2h^2\right)\right)}{4\varepsilon^2h^2-\sinh(4\varepsilon^2h^2)}$
τ_3	$-\frac{1}{120}h^4u^{(V)}(x_0)-\frac{1}{6}\varepsilon^2h^4u^{(III)}(x_0)-\frac{1}{2}\varepsilon^4h^4u'(x_0)+O\left(h^6P_3(\varepsilon^2)\right)$

A.3 RBF-HFD formulas

A.3.1 First derivative

Table A.9 shows the exact values of the weights and the corresponding local truncation error for RBF-HFD formulas to approximate the first derivative in 1D using the three node stencil



(A.1)

where the double circles represent the m-nodes which expand compactly the stencil centered at x_0 (see RBF-HFD formulation in Section 1.4).

Figure A.4 shows the corresponding numerical error (solid line) and compares it with the approximate error given by the formula in Table A.9 (dashed line). Notice that the agreement is excellent up to the point where the linear system to numerically compute the weights (1.24) becomes ill-conditioned and round-off errors deteriorate the accuracy of the numerical solution. The left part of Figure A.4 shows the absolute value of the error as a function of the shape parameter for $h = 0.05$. The accuracy increases with decreasing ε.

It has been shown that RBF-FD formulas approach standard finite difference formulas in the limit $\varepsilon \to 0$ [22]. Wright and Fornberg [109] studied RBF-HFD formulas and concluded that although there are not similar rigorous results for RBF-HFD formulas in the limit $\varepsilon \to 0$, they expected similar results to hold. In fact, taking the limit $\varepsilon \to 0$ in the weights given in Table A.9 results in $\alpha_{\pm 1} = \pm 3/4$, $\omega_{\pm 1} = -1/4$, which agrees with the results in

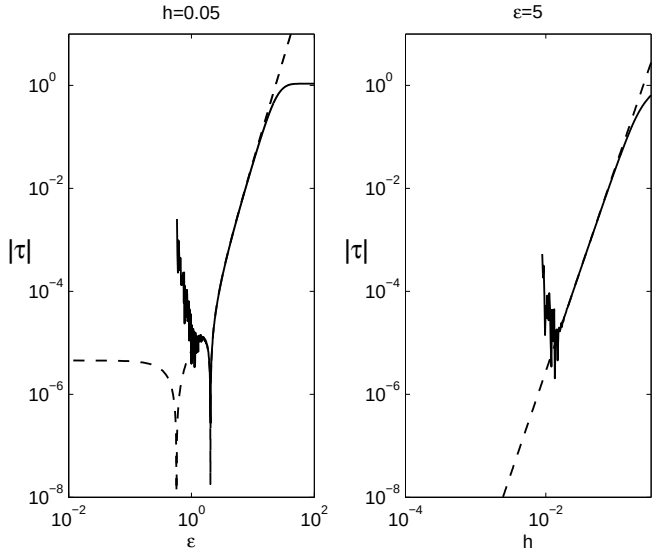


Figure A.4: Local truncation error (τ) for the RBF-HFD first derivative as function of ε (left side) and h (right side) using the stencils of Figure A.1. Solid line: local truncation error computed solving numerically (1.24). Dashed line: leading order formula of the error given in Table A.9.

Table 3, page 538 of [17].

It can be clearly observed that there are two distinct values of ε for which the error is zero to leading order. One of them occurs before the appearance of ill-conditioning and is accurately predicted by the approximate error formula. The other occurs in the region of ill-conditioning and, therefore, can not be seen with the numerical results. The right part of Figure A.4 shows the absolute value of the error as a function of the inter-nodal distance h for $\varepsilon = 5$. Notice that the error behaves as $O(h^4)$ in agreement with the formula in Table A.9.

Notice that the RBF-HFD formula with three nodes contains five weights (three of them independent). Thus, it should be compared to the RBF-FD formula for five nodes, which also contains five weights (three of them independent). Both have the same error dependence with h and with ε, although the RBF-HFD three nodes formula appears to be slightly more accurate than the RBF-FD five nodes formula. In fact, as $\varepsilon \to 0$ the RBF-HFD local truncation error approaches $-(1/120)h^4 u^{(5)}$ while the corresponding RBF-FD formula approaches $-(1/30)h^4 u^{(5)}$.

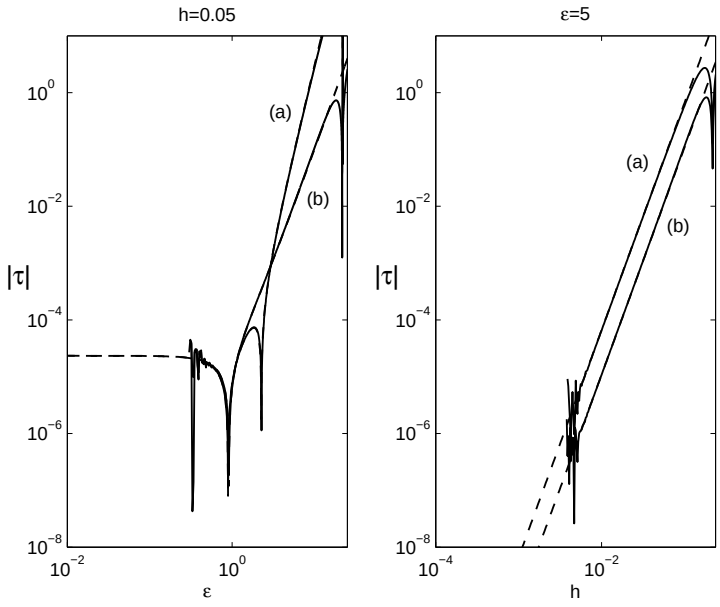


Figure A.5: Same as Figure A.4 but for the RBF-HFD second derivative: (a) formulation (1.24) non exact for constants, and (b) formulation (1.26) exact for constants.

A.3.2 Second derivative

Table A.10 shows the weights and the corresponding local truncation errors to approximate the second derivative in 1D using the three node stencil shown in Figure A.1. Results are shown both for the formulation which is not exact for constants (1.24) and for the formulation which is exact for constants (1.26). In this last case, only the series expansions of the coefficients in the limit $\varepsilon h \ll 1$ are included. It should be pointed out, that in the limit $\varepsilon \to 0$ the weights given in Table A.10; $\alpha_0 = -12/(5\,h^2)$, $\alpha_{\pm 1} = 6/(5\,h^2)$, $\omega_{\pm 1} = -1/10$, coincide with the results obtained with standard finite difference formulas (see Table 3, page 538 of [17]).

Figure A.5 shows the numerical error (solid line) in the approximation of the second derivative with three equispaced nodes, using both the formulation which is non exact for constants (1.24) and the formulation which is exact for constants (1.26). The numerical results are compared with the approximate error given by the formulas in Table A.10 (dashed line). There is an excellent agreement between the two results.

The left part of Figure A.5 shows the absolute value of the error as a function of the shape parameter for $h = 0.05$, and the right part shows the

absolute value of the error as a function of the inter-nodal distance h for $\varepsilon = 5$. Notice, that in the case of the formulation which is not exact for constants, there are two values of ε for which the error is zero to leading order; $\varepsilon^* \approx 2.2674$ and $\varepsilon^* \approx 0.8922$. For the formulation which is exact for constants there is only one value, $\varepsilon^* \approx 0.9129$, for which the error is zero to leading order. Similarly to what happened with the first derivative, the error dependence on h and ε of these RBF-HFD formulas using three nodes (three independent weights), equals the corresponding error dependence of RBF-FD formulas using five nodes (three independent weights). However, in the limit $\varepsilon \to 0$ the accuracy of the RBF-HFD ($\tau_3 \approx -(1/200)h^4 u^{(6)}$) appears to be slightly better than RBF-FD ($\tau_3 \approx -(1/90)h^4 u^{(6)}$).

Table A.10: RBF-HFD second derivative

Three nodes: non exact for constants	
α_0	$-\frac{2\varepsilon^2\left(-16\varepsilon^2h^2+e^{8\varepsilon^2h^2}+2e^{4\varepsilon^2h^2}\left(1-4\varepsilon^2h^2\right)^2+e^{6\varepsilon^2h^2}\left(8\varepsilon^6h^6-20\varepsilon^4h^4+8\varepsilon^2h^2-2\right)+e^{2\varepsilon^2h^2}\left(72\varepsilon^6h^6-52\varepsilon^4h^4+24\varepsilon^2h^2-2\right)+1\right)}{-16\varepsilon^2h^2+e^{8\varepsilon^2h^2}+2e^{4\varepsilon^2h^2}(1-4\varepsilon^2h^2)^2+e^{2\varepsilon^2h^2}(-36\varepsilon^4h^4+32\varepsilon^2h^2-2)-2e^{6\varepsilon^2h^2}(2\varepsilon^4h^4+1)+1}$
$\alpha_{\pm1}$	$\frac{2\varepsilon^2e^{3\varepsilon^2h^2}\left(48\varepsilon^6h^6-6\varepsilon^4h^4-10\varepsilon^2h^2+e^{4\varepsilon^2h^2}\left(2\varepsilon^4h^4-2\varepsilon^2h^2+1\right)-2e^{2\varepsilon^2h^2}\left(8\varepsilon^4h^4-6\varepsilon^2h^2+1\right)+1\right)}{-16\varepsilon^2h^2+e^{8\varepsilon^2h^2}+2e^{4\varepsilon^2h^2}(1-4\varepsilon^2h^2)^2+e^{2\varepsilon^2h^2}(-36\varepsilon^4h^4+32\varepsilon^2h^2-2)-2e^{6\varepsilon^2h^2}(2\varepsilon^4h^4+1)+1}$
$\omega_{\pm1}$	$\frac{e^{3\varepsilon^2h^2}\left(-6\varepsilon^4h^4-4\varepsilon^2h^2+e^{2\varepsilon^2h^2}\left(8\varepsilon^2h^2-2\right)+e^{4\varepsilon^2h^2}\left(2\varepsilon^4h^4-4\varepsilon^2h^2+1\right)+1\right)}{-16\varepsilon^2h^2+e^{8\varepsilon^2h^2}+2e^{4\varepsilon^2h^2}(1-4\varepsilon^2h^2)^2+e^{2\varepsilon^2h^2}(-36\varepsilon^4h^4+32\varepsilon^2h^2-2)-2e^{6\varepsilon^2h^2}(2\varepsilon^4h^4+1)+1}$
τ_3	$-\frac{1}{200}h^4u^{(VI)}(x_0)-\frac{3}{20}\varepsilon^2h^4u^{(IV)}(x_0)-\frac{9}{10}\varepsilon^4h^4u''(x_0)-\frac{3}{5}\varepsilon^6h^4u(x_0)+O\left(h^6P_4(\varepsilon^2)\right)$
Three nodes: exact for constants	
α_0	$-\frac{12}{5h^2}-\frac{84\varepsilon^2}{125}+\frac{3021h^2\varepsilon^4}{3125}+O\left(h^4\varepsilon^6\right)$
$\alpha_{\pm1}$	$\frac{6}{5h^2}+\frac{42\varepsilon^2}{125}-\frac{3021h^2\varepsilon^4}{6250}+O\left(h^4\varepsilon^6\right)$
$\omega_{\pm1}$	$-\frac{1}{10}-\frac{21h^2\varepsilon^2}{125}-\frac{677h^4\varepsilon^4}{6250}+O\left(h^4\varepsilon^6\right)$
τ_3	$-\frac{1}{200}h^4u^{(VI)}(x_0)-\frac{7}{50}\varepsilon^2h^4u^{(IV)}(x_0)-\frac{7}{10}\varepsilon^4h^4u''(x_0)+O\left(h^6P_3(\varepsilon^2)\right)$

A.4 Conclusions

In this appendix, we derive analytical expressions for the weights of RBF-FD and RBF-HFD formulas for first and second derivatives in 1D, and for the Laplacian in 2D using Gaussians as RBFs. Results are presented for 3, 5, 7 and 9 nodes in the case of RBF-FD formulas in 1D, and for 5 and 9 nodes in the case of RBF-FD formulas in 2D. For the case of RBF-HFD formulas we compute the weights for first and second order derivatives, using three equispaced nodes only. These weights are then used to derive analytical expressions for the leading order approximations to the local error in powers of the inter-nodal distance h.

As in the multiquadric case from Chapter 2, there is a range of values of the shape parameter for which RBF-FD formulas and RBF-HFD formulas are significantly more accurate than the corresponding conventional finite difference formulas. In fact, very often there is an optimal value of the shape parameter ε^* for which the error is zero to leading order. This value can be easily estimated from the analytical expressions. As we show in Chapter 2, it is independent of the inter-nodal distance and only depends to leading order on the value of the function and its derivatives at the node. Therefore, the results presented in this appendix can be used to efficiently select an optimal value of the shape parameter as it is proposed in Chapters 3 and 4.

Publications

The work in this thesis has resulted in the following papers:

1. V. Bayona and M. Kindelan, "Propagation of premixed laminar flames in 3D narrow open ducts using RBF-generated finite differences", *Combustion theory and modelling*, accepted (2013).

2. S. Simonenko, V. Bayona and M. Kindelan, "Optimal Shape Parameter for the solution of Elastostatic Problems with the RBF Method", *Journal of Engineering Mathematics*, accepted (2012).

3. V. Bayona, M. Moscoso and M. Kindelan, "Gaussian RBF-FD weights and its corresponding local truncation errors", *Engineering Analysis with Boundary Elements*, 36 (9) 1361-1369 (2012).

4. V. Bayona, M. Moscoso and M. Kindelan, "Optimal *variable* shape parameter for multiquadric based RBF-FD method", *J. Comput. Phys.* 231 (6), 2466-2481 (2012).

5. V. Bayona, M. Moscoso and M. Kindelan, "Optimal *constant* shape parameter for multiquadric based RBF-FD method", *J. Comput. Phys.* 230 (19), 7384-7399 (2011).

6. V. Bayona, M. Moscoso and M. Kindelan, "RBF-FD formulas and convergence properties", *J. Comput. Phys.* 229 (22), 8281-8295 (2010).

Resumen en Español

En el campo de los *métodos sin malla*, los métodos RBF (Funciones de Base Radial, del inglés *Radial Basis Function*) se han convertido en una alternativa real para la interpolación de funciones o la solución de ecuaciones en derivadas parciales (EDPs) en dominios irregulares. La dependencia de la distancia entre nodos los convierte en métodos simples y fáciles de implementar, independientemente de la dimensión y/o geometría del dominio. Existen dos formulaciones distintas para la resolución de EDPs: el *método RBF global* y el *método RBF local*.

En el método global, la solución aproximada se calcula en el espacio funcional de las RBFs. Las coordenadas de la solución en ese espacio se obtienen mediante colocación. Dicha formulación da lugar a matrices de diferenciación densas que aproximan el operador con convergencia espectral. La principal desventaja es que, a medida que el número total de nodos aumenta, el número de condición de la matriz de colocación tambien aumenta, lo que restringe su utilidad en problemas prácticos.

Para salvar algunos de los problemas del método RBF global, varios autores han propuesto de manera independiente el método RBF local, al que le han dado diferentes nombres: Shu *et al.* [95] *local multiquadric-based differential quadrature*, Tolstykh *et al.* [100] *RBF in a "finite difference mode"*, Wright [108] *RBF-generated finite differences*. Este método puede considerarse una generalización natural del método de diferencias finitas: dado un operador diferencial y un conjunto de nodos, el método RBF local aproxima la acción de un operador diferencial en un punto como una combinación lineal ponderada de la función buscada en un conjunto de nodos cercanos, que forman el *stencil*. Sin embargo, mientras que en diferencias finitas los pesos se calculan por medio de una interpolación polinómica, en el método RBF local se calculan a través de una función interpolante RBF. Dado que ambos métodos son conceptualmente idénticos, el método RBF local se denomina comunmente *RBF-generated finite differences* (RBF-FD), tal y como propuso Wright en [108].

A diferencia del método global, el método local no ha recibido mucha

atención a lo largo de los años ya que no presenta convergencia espectral. Sin embargo, la capacidad para resolver EDPs en dominios irregulares mediante matrices de diferenciación dispersas, junto con la posibilidad de aproximar el operador a alto orden de precisión, lo sitúan a la vanguardia de los métodos sin malla.

En la presente tesis aportamos un estudio sobre algunos de los aspectos del método que a nuestro juicio no han sido suficientemente desarrollados. Con este fin, revisamos el estado del método en el primer capítulo, donde explicamos los conceptos básicos a los que haremos referencia a lo largo de la tesis y demostramos la equivalencia formal entre el método global y local, que parece no ser del todo conocida por la comunidad RBF. El resto de la tesis se divide en dos partes.

En la primera parte abordamos el problema de la convergencia del método. En el capítulo 2 y Apéndice A, obtenemos fórmulas para el error local de truncamiento en función de la distancia internodal h, el parámetro de forma ε y el tamaño del stencil n. Estas fórmulas demuestran la evidencia experimental de que la precisión del método depende del parámetro de forma. A medida que $\varepsilon \to 0$, el error decrece hasta que el parámetro de forma óptimo se alcanza. Dicho valor es el que obtiene resultados más precisos. Para valores más pequeños, la función de interpolación RBF converge a una interpolación polinómica, el error se satura y los resultados terminan coincidiendo con métodos basados en interpolación polinómica. El principal resultado de nuestro estudio es la obtención de fórmulas explícitas para el parámetro de forma óptimo, con el que se obtienen resultados mucho más precisos que con diferencias finitas. A primer orden, dicho valor es independiente de las distancia internodal y sólo depende de la función y sus derivadas.

Basándonos en las fórmulas analíticas del error local de truncamiento, abordamos el problema de cómo elegir el parámetro de forma óptimo para la resolución de EDP's. Así, presentamos dos algoritmos en los Capítulos 3 y 4. En el primero, el parámetro de forma se calcula minimizando la norma infinita del error global. A través de varios experimentos numéricos, demostramos que la precisión de la solución puede mejorar en uno o dos órdenes de magnitud con respecto a diferencias finitas. En el segundo algoritmo planteamos un parámetro de forma dependiente de la posición, que se obtiene minimizando el error local de truncamiento en cada nodo del dominio. Para asegurar la existencia de un parámetro de forma óptimo en todos los nodos proponemos el uso de MQ generalizadas como RBFs. De esta forma, se obtienen significantes mejoras en la precisión con respecto al primer algoritmo.

En la segunda parte de la tesis exploramos la aplicabilidad del método

en la resolución de problemas prácticos. Así, en el Capítulo 5 resolvemos algunos de los problemas lineales clásicos de la elasticidad. En él, aplicamos los algoritmos anteriores y mostramos que es posible mejorar la precisión significativamente eligiendo el parámetro de forma correctamente.

El orden de convergencia de la aproximación se puede mejorar aumentanado el número de nodos en el *stencil*. Sin embargo, sólo disponemos de fórmulas analíticas para el cálculo del parámetro de forma óptimo para *stencils* relativamente pequeños. Ampliar los resultados de modo que en estos casos dispongamos también de dichas fórmulas, conlleva resolver sistemas de ecuaciones mayores que los ya resueltos. Desde el punto de vista analítico, esto entraña ciertas dificultades que dejamos para un futuro trabajo.

En estos casos, parece más conveniente utilizar el procedimiento propuesto en [38, 88]. En él, el parámetro de forma se calcula de modo que el número de condicionamiento de la matriz de colocación se encuentre acotado en la región en la que el sistema está todavía bien condicionado y el error es cercano a su valor mínimo, evitando así el problema del mal condicionamiento. En la Sección 1.6 del Capítulo 1 analizamos este procedimiento a través de un ejemplo numérico. También lo aplicamos a la resolución de problemas no lineales en dominios irregulares. Así, en el Capítulo 6 resolvemos un problema tridimensional de combustión donde utilizamos *stencils* formados por un gran número de nodos, dando lugar a aproximaciones de alto orden. El problema consiste en dos ecuaciones de convección-difusión acopladas a través de un término de reacción no lineal. Los resultados obtenidos son bastante satisfactorios, lo que nos lleva a implementar un modelo matemático para el funcionamiento de un micromotor Wankel en el Capítulo 7. En este modelo ideal, se asume que el campo de velocidades del combustible es independiente de la combustión y puede ser determinado a priori utilizando las ecuaciones estacionarias de Navier-Stokes. De esta forma, una vez calculado el campo de velocidades, utilizamos el modelo del Capítulo 6 para simular la combustión dentro del motor, donde el término convectivo incluye ya dicho campo de velocidades.

Bibliography

[1] C.E. Augarde and A.J. Deeks. The use of Timoshenko's exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design*, 44(9-10):595–601, 2008.

[2] R.K. Beatson, W.A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal on Scientific Computing*, 22(5):1717–1740, 2001.

[3] R.K. Beatson and G.N. Newsam. Fast evaluation of radial basis functions: I. *Computers & Mathematics with Applications*, 24(12):7–19, 1992.

[4] R.K. Beatson and G.N. Newsam. Fast evaluation of radial basis functions: Moment-based methods. *Siam Journal on Scientific Computing*, 19(5):1428–1449, 1998.

[5] E.F. Bollig, N. Flyer, and G. Erlebacher. Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs. *J. Comput. Phys.*, 231(21):7133–7151, 2012.

[6] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Courier Dover Publications, 2001.

[7] J.P. Boyd and L. Wang. Truncated gaussian RBF differences are always inferior to finite differences of the same stencil width. *Communications in Computational Physics*, 5(1):42–60, 2009.

[8] M.D. Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[9] R.E. Carlson and T.A. Foley. The parameter R^2 in multiquadric interpolation. *Computers and Mathematics with Applications*, 21(9):29–42, 1991.

[10] G.F. Carrier, F.E. Fendell, and P.S. Feldman. Laminar flame propagation/quench for a parallel-wall duct. *Symposium (International) on Combustion*, 20(1):67–74, 1985.

[11] G. Chandhini and Y.V.S.S. Sanyasiraju. Local RBF-FD solutions for steady convection-diffusion problems. *International Journal for Numerical Methods in Engineering*, 72(3):352–378, 2007.

[12] A.H.-D. Cheng. Multiquadric and its shape parameter - a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering Analysis with Boundary Elements*, 36(2):220–239, 2012.

[13] J. B. Cherrie, R. K. Beatson, and G. N. Newsam. Fast evaluation of radial basis functions: Methods for generalized multiquadrics in $\mathcal{R}^n$. *SIAM Journal on Scientific Computing*, 23(5):1549–1571, 2002.

[14] P.P. Chinchapatnam. *Radial basis function based meshless methods for fluid flow problems*. PhD thesis, University of Southampton, 2006.

[15] H.K. Ching. *Solution of Linear Elastostatic and Elastodynamic Plane Problems by the Meshless Local Petrov-Galerkin Method*. PhD thesis, Virginia Polytechnic Institute and State University, 2002.

[16] A. J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[17] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer Verlag, Berlin, 1966.

[18] P.C. Curtis. N-parameter families and best approximation. *Pacific J. Math*, 9(4):1013–1027, 1959.

[19] O. Davydov and D.T. Oanh. Adaptive meshless centres and RBF stencils for poisson equation. *Journal of Computational Physics*, 230(2):287–304, 2011.

[20] H. Ding, C. Shu, and D.B. Tang. Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments. *International Journal for Numerical Methods in Engineering*, 63(11):1513–1529, 2005.

[21] H. Ding, C. Shu, K.S. Yeo, and D. Xu. Numerical computation of three-dimensional incompressible viscous flows in the primitive variable

form by local multiquadric differential quadrature method. *Computer methods in applied mechanics and engineering*, 195(7):516–533, 2006.

[22] T.A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Computers and Mathematics with Applications*, 43(3-5):413–422, 2002.

[23] T.A. Driscoll and A.R.H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Computers and Mathematics with Applications*, 53(6):927–939, 2007.

[24] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[25] G.E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In *Proceedings of Chamonix*, pages 1–8. Citeseer, 1996.

[26] G.E. Fasshauer. Solving differential equations with radial basis functions: multilevel methods and smoothing. *Advances in Computational Mathematics*, 11(2-3):139–159, 1999.

[27] G.E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. Interdisciplinary mathematical sciences. World Scientific, 2007.

[28] G.E. Fasshauer and M.J. McCourt. Stable evaluation of gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):A737–A762, 2012.

[29] G.E. Fasshauer and J.G. Zhang. On choosing "optimal" shape parameters for RBF approximation. *Numerical Algorithms*, 45(1-4):345–368, 2007.

[30] A.J.M. Ferreira. A formulation of the multiquadric radial basis function method for the analysis of laminated composite plates. *Composite Structures*, 59(3):385–392, 2003.

[31] A.J.M. Ferreira. Thick composite beam analysis using a global meshless approximation based on radial basis functions. *Mechanics of Advanced Materials and Structures*, 10(3):271–284, 2003.

[32] A.J.M. Ferreira and G.E. Fasshauer. Computation of natural frequencies of shear deformable beams and plates by an RBF-pseudospectral method. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):134–146, 2006.

[33] A.J.M. Ferreira and G.E. Fasshauer. Analysis of natural frequencies of composite plates by an RBF-pseudospectral method. *Composite Structures*, 79(2):202–210, 2007.

[34] A.J.M. Ferreira, G.E. Fasshauer, R.C. Batra, and J.D. Rodrigues. Static deformations and vibration analysis of composite and sandwich plates using a layerwise theory and RBF-PS discretizations with optimal shape parameter. *Composite Structures*, 86(4):328–343, 2008.

[35] A.J.M. Ferreira, C.M.C. Roque, R.M.N. Jorge, and E.J. Kansa. Static deformations and vibration analysis of composite and sandwich plates using a layerwise theory and multiquadrics discretizations. *Engineering Analysis with Boundary Elements*, 29(12):1104–1114, 2005.

[36] A.J.M. Ferreira, C.M.C. Roque, and P.A.L.S. Martins. Radial basis functions and higher-order shear deformation theories in the analysis of laminated composite beams and plates. *Composite Structures*, 66(1-4):287–293, 2004.

[37] M.S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1):65–78, 1996.

[38] N. Flyer, E. Lehto, S. Blaise, G.B. Wright, and A. St-Cyr. A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere. *Journal of Computational Physics*, 231(11):4078–4095, 2012.

[39] B. Fornberg and N. Flyer. Radial basis functions: Developments and applications. SIAM Press, Philadelphia, PA, due Summer 2012.

[40] B. Fornberg and E. Lehto. Stabilization of RBF-generated finite difference methods for convective PDEs. *Journal of Computational Physics*, 230(6):2270–2285, 2011.

[41] B. Fornberg, E. Lehto, and C. Powell. Stable calculation of gaussian-based RBF-FD stencils. *Computers & Mathematics with Applications*, 2012.

[42] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2007.

[43] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers and Mathematics with Applications*, 48(5-6):853–867, 2004.

[44] B. Fornberg, G. Wright, and E. Larsson. Some observations regarding interpolants in the limit of flat radial basis functions. *Computers and Mathematics with Applications*, 47(1):37–55, 2004.

[45] R. Franke. A critical comparison of some methods for interpolation of scattered data. Technical report, DTIC Document, 1979.

[46] R. Franke. Scattered data interpolation: Tests of some methods. *Math. Comput.*, 38(157):181–200, 1982.

[47] J. Fröhlich and J. Lang. Two-dimensional cascadic finite element computations of combustion problems. *Computer Methods in Applied Mechanics and Engineering*, 158(3-4):255–267, 1998.

[48] S. Galant. Multiplicity and stability of burner stabilized premixed laminar flames: A general proof. In *Symposium (International) on Combustion*, volume 18, pages 1343–1353. Elsevier, 1981.

[49] J.H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[50] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.*, 76(8):1905–1915, 1971.

[51] R.F. Heinemann, K.A. Overholser, and G. W. Reddien. Multiplicity and stability of premixed laminar flames: An application of bifurcation theory. *Chemical Engineering Science*, 34(6):833–840, 1979.

[52] Y.C. Hon and R. Schaback. On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation*, 119(2):177–186, 2001.

[53] C.S. Huang, C.F. Lee, and A.H.D. Cheng. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. *Engineering Analysis with Boundary Elements*, 31(7):614–623, 2007.

[54] C.S. Huang, H.D. Yen, and A.H.D. Cheng. On the increasingly flat radial basis function and optimal shape parameter for the solution of elliptic PDEs. *Engineering Analysis with Boundary Elements*, 34(9):802–809, 2010.

[55] E.J. Kansa. Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, 19(8-9):127–145, 1990.

[56] E.J. Kansa. Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, 19(8-9):147–161, 1990.

[57] E.J. Kansa and R.E. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers and Mathematics with Applications*, 24(12):99–120, 1992.

[58] E.J. Kansa and Y.C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Computers and Mathematics with Applications*, 39(7-8):123–137, 2000.

[59] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[60] V.N. Kurdyumov and E. Fernández-Tarrazo. Lewis number effect on the propagation of premixed laminar flames in narrow open ducts. *Combustion and Flame*, 128(4):382–394, 2002.

[61] V.N. Kurdyumov, E. Fernández-Tarrazo, and A. Liñán. Flame flashback and propagation of premixed flames near a wall. *Symposium (International) on Combustion*, 28(2):1883–1889, 2000.

[62] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[63] B. Larrouturou. Adaptive numerical simulation of premixed flame propagation. *Numerical Modeling in Combustion*, pages 133–278, 1993.

[64] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers and Mathematics with Applications*, 46(5-6):891–902, 2003.

[65] E. Larsson and B. Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Computers and Mathematics with Applications*, 49(1):103–130, 2005.

[66] D.G. Lasseigne, T.L. Jackson, and L. Jameson. Stability of freely propagating flames revisited. *Combustion Theory and Modelling*, 3(4):591–611, 1999.

[67] E. Lehto. *High Order Local Radial Basis Function Methods for Atmospheric Flow Simulations*. PhD thesis, Uppsala UniversityUppsala University, Division of Scientific Computing, Numerical Analysis, 2012.

[68] V.M.A. Leitão. A meshless method for kirchhoff plate bending problems. *International Journal for Numerical Methods in Engineering*, 52(10):1107–1130, 2001.

[69] S.K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42, 1992.

[70] J. Li and Y.C. Hon. Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations*, 20(3):450–462, 2004.

[71] K.M. Liew, X.L. Chen, and J.N. Reddy. Mesh-free radial basis function method for buckling analysis of non-uniformly loaded arbitrarily shaped shear deformable plates. *Computer Methods in Applied Mechanics and Engineering*, 193(3-5):205–224, 2004.

[72] L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Mathematical and Computer modelling*, 40(13):1413–1427, 2004.

[73] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11(1-2):83–95, 1980.

[74] G.R. Liu and Y.T. Gu. A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. *Journal of Sound and Vibration*, 246(1):29–46, 2001.

[75] G.R. Liu, G.Y. Zhang, Y.T. Gu, and Y.Y. Wang. A meshfree radial point interpolation method (RPIM) for three-dimensional solids. *Computational Mechanics*, 36(6):421–430, 2005.

[76] W.R. Madych. Miscellaneous error bounds for multiquadric and related interpolators. *Computers & Mathematics with Applications*, 24(12):121–138, 1992.

[77] W.R. Madych and S.A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70(1):94–114, 1992.

[78] C. Mairhuber. On haar's theorem concerning chebyshev approximation problems having unique solutions. *Proc. Amer. Math. Soc.*, 7:609–615, 1956.

[79] C.A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

[80] F.J. Narcowich and J.D. Ward. Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices. *Journal of Approximation Theory*, 69(1):84–109, 1992.

[81] N. Peters and J. Warnatz. *Numerical methods in laminar flame propagation*, volume 6. Notes on Numerical Fluid Mechanics. Vieweg & Sohn, 1982.

[82] J. I. Ramos. *Finite element methods for one-dimensional flame propagation problems.* Numerical Modeling in Combustion. T.J. Chung, Taylor & Francis, 1993.

[83] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3):193–210, 1999.

[84] C.M.C. Roque, D. Cunha, C. Shu, and A.J.M. Ferreira. A local radial basis functions finite differences technique for the analysis of composite plates. *Engineering Analysis with Boundary Elements*, 35(3):363–374, 2011.

[85] C.M.C. Roque, A.J.M. Ferreira, and R.M.N. Jorge. A radial basis function approach for the free vibration analysis of functionally graded plates using a refined theory. *Journal of Sound and Vibration*, 300(3-5):1048–1070, 2007.

[86] R. L. Sani and P. M. Gresho. Resume and remarks on the open boundary condition minisymposium. *International Journal for Numerical Methods in Fluids*, 18(10):983–1008, 1994.

[87] Y.V.S.S. Sanyasiraju and G. Chandhini. Local radial basis function based gridfree scheme for unsteady incompressible viscous flows. *Journal of Computational Physics*, 227(20):8922–8948, 2008.

[88] S.A. Sarra. A local radial basis function method for advection–diffusion–reaction equations on complexly shaped domains. *Applied Mathematics and Computation*, 218:9853–9865, 2012.

[89] S.A. Sarra and E.J. Kansa. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. *Tech Science Press*, 1:2, 2010.

[90] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Adv. Comput. Math.*, 3:251–264, 1995.

[91] R. Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constructive approximation*, 21(3):293–317, 2005.

[92] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39(4):pp. 811–841, 1938.

[93] B. Seibold. *M-Matrices in meshless finite difference methods*. PhD thesis, Department of Mathematics, University of Kaiserslautern, 2006.

[94] B. Seibold. Minimal positive stencils in meshfree finite difference methods for the poisson equation. *Computer Methods in Applied Mechanics and Engineering*, 198(3-4):592–601, 2008.

[95] C. Shu, H. Ding, and K.S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(7-8):941–954, 2003.

[96] S. B. Sprague, S.W. Park, D. C. Walther, and A. P. Pisano. Development and characterisation of small-scale rotary engines. *International Journal of Alternative Propulsion*, 1(2):275–293, 2007.

[97] S. Tanaka, T. Yamada, S. Sugimoto, J.F. Li, and M. Esashi. Silicon nitride ceramic-based two-dimensional microcombustor. *Journal of Micromechanics and Microengineering*, 13(3):502, 2003.

[98] S.P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill, New York, 1951.

[99] A.I. Tolstykh. On using rbf-based differencing formulas for unstructured and mixed structured-unstructured grid calculations. In *Proceedings of the 16th IMACS World Congress, Lausanne*, 2000.

[100] A.I. Tolstykh and D.A. Shirobokov. On using radial basis functions in a "finite difference mode" with applications to elasticity problems. *Computational Mechanics*, 33(1):68–79, 2003.

[101] J. Vican, B.F. Gajdeczko, F.L. Dryer, D.L. Milius, I.A. Aksay, and R.A. Yetter. Development of a microreactor as a thermal source for microelectromechanical systems power generation. *Proceedings of the Combustion Institute*, 29(1):909–916, 2002.

[102] D.C. Walther and J. Ahn. Advances and challenges in the development of power-generation systems at small scales. *Progress in Energy and Combustion Science*, 37(5):583–610, 2011.

[103] J.G. Wang and G.R. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54(11):1623–1648, 2002.

[104] F.J. Weinberg, D.M. Rowe, G. Min, and P.D. Ronney. On thermoelectric power conversion from heat recirculating combustion systems. *Proceedings of the Combustion Institute*, 29(1):941–947, 2002.

[105] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.

[106] H. Wendland. *Scattered data approximation*, volume 17. Cambridge University Press Cambridge, 2005.

[107] J. Wertz, E.J. Kansa, and L. Ling. The role of the multiquadric shape parameters in solving elliptic partial differential equations. *Computers and Mathematics with Applications*, 51(8 SPEC. ISS.):1335–1348, 2006.

[108] G. B. Wright. *Radial Basis Function Interpolation: Numerical and Analytical Developments*. PhD thesis, Department of Applied Mathematics, University of Colorado, 2003.

[109] G.B. Wright and B. Fornberg. Scattered node compact finite difference-type formulas generated from radial basis functions. *Journal of Computational Physics*, 212(1):99–123, 2006.

[110] Z. Wu. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8(2):1–10, 1992.

[111] X. Zhang, K.Z. Song, M.W. Lu, and X. Liu. Meshless methods based on collocation with radial basis functions. *Computational Mechanics*, 26(4):333–343, 2000.

[112] X. Zhou, Y.C. Hon, and J. Li. Overlapping domain decomposition method by radial basis functions. *Applied Numerical Mathematics*, 44(1):241–255, 2003.